z/OS

IBM

DFSMS Advanced Copy Services

Version 2 Release 2

SC23-6847-02

Note

Before using this information and the product it supports, read the information in "Notices" on page 817.

This edition applies to Version 2 Release 2 of z/OS (5650-ZOS) and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Chapter 13. Setting up the Peer-to-Peer Remote copy environment 281

Chapter 14. Peer-to-Peer Remote copy command descriptions 291

Chapter 15. Managing Peer-to-Peer Remote Copy operations 321

Chapter 16. Peer-to-Peer Remote Copy data recovery operations. . . . 367

Figures

Tables

About this document

The purpose of this publication is to help you understand and use IBM® Advanced Copy Services functions. It describes three "dynamic" copy functions and several "point-in-time" copy functions. These functions provide backup and recovery of data should a disaster occur to your data center. The dynamic functions are peer-to-peer remote copy, extended remote copy, and coupled extended remote copy, and are known collectively as remote copy. FlashCopy®, SnapShot, and Concurrent Copy are the point-in-time copy functions.

For information about the accessibility features of z/OS®, for users who have a physical disability, see Appendix E, "Accessibility," on page 813.

Required product knowledge

To use this document effectively, you should be familiar with the following information:

- Programming, especially programming with TSO commands
- Current disaster recovery and workload migration procedures at your location

The person using this information must understand the effects of changing system parameters. For some cases, such as adjustments to the PARMLIB parameters, it is recommended that you contact an IBM representative before you make any adjustments.

z/OS information

This information explains how z/OS references information in other documents and on the web.

When possible, this information uses cross document links that go directly to the topic in reference using shortened versions of the document title. For complete titles and order numbers of the documents for all products that are part of z/OS, see *z/OS Information Roadmap*.

To find the complete z/OS library, go to IBM Knowledge Center (http://www.ibm.com/support/knowledgecenter/SSLTBW/welcome).

How to send your comments to IBM

We appreciate your input on this publication. Feel free to comment on the clarity, accuracy, and completeness of the information or provide any other feedback that you have.

Use one of the following methods to send your comments:

1. Send an email to mhvrcfs@us.ibm.com.
2. Send an email from the "Contact us" web page for z/OS (http://www.ibm.com/systems/z/os/zos/webqs.html).

Include the following information:

- Your name and address.
- Your email address.
- Your telephone or fax number.
- The publication title and order number:
  z/OS V2R2 DFSMS Advanced Copy Services
  SC23-6847-02
- The topic and page number that is related to your comment.
- The text of your comment.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute the comments in any way appropriate without incurring any obligation to you.

IBM or any other organizations use the personal information that you supply to contact you only about the issues that you submit.

If you have a technical problem

Do not use the feedback methods that are listed for sending comments. Instead, take one of the following actions:

- Contact your IBM service representative.
- Call IBM technical support.
- Visit the IBM Support Portal at z/OS Support Portal (http://www-947.ibm.com/systems/support/z/zos/).

Summary of changes

This information includes terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations for the current edition are indicated by a vertical line to the left of the change.

Summary of changes for z/OS Version 2 Release 2 (V2R2)

The following changes are made for z/OS Version 2 Release 2 (V2R2).

Changed

The XRECOVER command now accepts CHECK and FORCE parameters, to control XRECOVER enhanced checking. For more information, refer to "XRECOVER–Recovering data on the recovery system" on page 100.

The XRECOVER request of the ANTRQST macro and ANTTREXX program now accept CHECK and FORCE. For more information, refer to "Subparameters for REQUEST=XRECOVER" on page 717 and "XRECOVER (XRC recover)" on page 803.

The PQUERY request of the ANTRQST macro and ANTTREXX program now accept QUIESCEDIO. For more information, refer to "Subparameters for REQUEST=PQUERY" on page 675 and "PQUERY (PPRC query)" on page 780.

The GMPStatus field of GMPSTAT can have a new value. For more information, refer to Table 57 on page 435.

Summary of changes for z/OS Version 2 Release 1 (V2R1) as updated February, 2015

The following changes are made for z/OS Version 2 Release 1 (V2R1) as updated February, 2015. Technical changes for this revision are indicated by a vertical line to the left of the change.

New

- Support for Multi-Target Mirror environments. For details, refer to "Choosing Multi-Target Mirror" on page 20.
- PSETCHAR request type, to set PPRC volume pair characteristics, including whether the indicated pair is to be used for a Preserve Mirror function in a Multi-Target Mirror configuration. For more information, refer to "PSETCHAR – set PPRC volume pair characteristics" on page 318, "Subparameters for REQUEST=PSETCHAR" on page 687, "PSETCHAR (PPRC set characteristics of volume pair)" on page 785, "Converting to Multi-Target Mirror" on page 275 and "Defining resource profiles in the RACF Facility class" on page 282.
- Message ANTP7026I, issued for a RQUERY DVCSTAT or STAT4C request in a Multi-Target environment. For more information, refer to "Querying Global Mirror summary output at device level" on page 436.
- Support for multiple Incremental FlashCopy targets, with Incremental FlashCopy Version 2. For more information, refer to "Incremental FlaschCopy" on page 484.

- Support for workload-based write pacing. For more information, refer to "Workload-based write pacing" on page 161, and the description of the new WorkloadWritePacing parameter in "ANTXIN00 parmlib parameters" on page 126.
- The XFEATURES parameter on the XQUERY STORAGECONTROL command, to show information related to workload-based write pacing. For more information, refer to "XQUERY–Querying a session" on page 90 and "XQUERY storage control XFEATURES report" on page 187.
- The XFEATURES parameter on XQUERY requests using ANTRQST or ANTTREXX. For more information, refer to Appendix C, "ANTRQST and ANTRQSTL macros – call to the system data mover API," on page 573 and Appendix D, "REXX support for the ANTRQST API," on page 745.

Changed

- The ACTION parameter on PPRC Establish functions now accepts MTFAILOVER, to establish a Multi-Target Mirror configuration. For more information, refer to "Adding a PPRC volume pair" on page 329, "Subparameters for REQUEST=PESTPAIR" on page 659, and "PESTPAIR (PPRC establish volume pair)" on page 772.
- The value for STATE in formatted output for CQUERY shows new values (MTIR and UNKNOWN) related to Multi-Target Mirror. In addition, the output for CQUERY VOLUME and CQUERY PATHS is expanded. For more information, refer to "Identifying Peer-to-Peer Remote Copy volume states" on page 322, "Fields in formatted output" on page 332, "CQUERY formatted output in a multi-target configuration" on page 341, and "CQUERY formatted output with the PATHS option in a Multi-Target configuration" on page 349.
- RVOLUME JOIN requests now accept MTVOLLIST and MTVOLRANGE, for Multi-Target Global Mirror sessions.For more information, refer to "RVOLUME – manage volumes for Global Mirror session" on page 424, "Subparameters for REQUEST=RVOLUME" on page 646, and "RVOLUME (Global Mirror manage volumes)" on page 792.
- The output for a RQUERY DVCSTAT request includes an additional ANTP7026I message in a Multi-Target environment. For more information, refer to "Querying Global Mirror summary output at device level" on page 436.
- In a PPRC configuration, you can now attach up to 16 recovery site logical subsystems to each primary site logical subsystem. For more information, refer to "Examining PPRC configuration options" on page 265 and Table 40 on page 288.
- "Incremental FlaschCopy" on page 484 is updated to reflect Incremental FlashCopy V2.
- "Optional parameters" on page 501 for the FCESTABL command is updated to indicate that PREFERRED is not supported in a Multi-Target Mirror configuration.
- "XQUERY volume pace report" on page 186 is updated for workload-based write pacing.
- The descriptions of ON for DVCBLOCK are changed to reflect workload-based write pacing, in "XADDPAIR–Adding volume pairs or utility volumes" on page 71 and "XSET–Changing session parameters" on page 102, "Adding a volume with the LOGPLUS option" on page 174, "Subparameters for REQUEST=XADD" on page 701, "Subparameters for REQUEST=XSET" on page 723, "XADD (XRC add volume pair)" on page 796, and "XSET (XRC session parameters)" on page 806.

Deleted

No content was removed from this information.

Summary of changes for z/OS Version 2 Release 1

See the following publications for all enhancements to z/OS Version 2 Release 1 (V2R1):

- *z/OS Migration*
- *z/OS Planning for Installation*
- *z/OS Summary of Message and Interface Changes*
- *z/OS Introduction and Release Guide*

Part 1. Advanced Copy Services Overview

This topic provides overviews to help you understand Advanced Copy Services.

Chapter 1. Introducing Advanced Copy Services

As businesses become more dependent on information technology to conduct their operations, the requirements for protecting their data, backing it up, and making it available, 24 hours a day and seven days a week are forever increasing. In the event of disaster or system failures, without a plan for recovery, loss of data can lead to significant financial cost or maybe total failure for businesses.

IBM Advanced Copy Services provides a combined hardware and software solution to help address these issues. Advanced Copy Services is a collection of functions that provides solutions to the complex challenges of disaster recovery, data migration, data duplication, and business continuance. Many of these functions run on IBM TotalStorage Enterprise Storage Server® (ESS). The ESS, designed for mid-range and high-end environments, gives you a high performance, continuous availability, high capacity storage subsystem that can be configured according to your requirements.

This chapter provides overviews to help you understand the functions of Advanced Copy Services.

In this topic

This topic introduces you to the functions of Advanced Copy Services. It includes the following sections:

Section . . .
"Overview of the Advanced Copy Services functions"
"Using the ESS Copy services Web interface" on page 7
"Combining copy services operations" on page 7

Overview of the Advanced Copy Services functions

Advanced Copy Services comprises the following functions and enhancements. *Mirror* functions provide a consistent point-in-time copy of data at the recovery site, while the data at the recovery site for *Copy* functions is not necessarily consistent.

- Metro Mirror, also known as (synchronous) Peer-to-Peer Remote Copy (PPRC)
- Global Mirror, also known as (asynchronous) Peer-to-Peer Remote Copy
- Global Copy, also known as PPRC-Extended Distance (PPRC-XD)
- Metro/Global Copy
- Global Mirror for System z®, also known as z/OS Global Mirror (zGM) and Extended Remote Copy (XRC), which includes Coupled Extended Remote Copy (CXRC)
- Metro/Global Mirror, also known as synchronous PPRC combined with Global Mirror
- Multi-Target Mirror (MTPPRC)
- FlashCopy
- Failover/Failback

- SnapShot
- Concurrent Copy (CC)

Table 1 shows how each of the Advanced Copy Services copy functions falls into one of two groups. Dynamic copy functions constantly update the secondary copy as applications make changes to the primary data source. Point-in-time copy functions provide an instantaneous copy, or view, of what the original data looked like at a specific point in time.

Table 1. Dynamic copy and point-in-time copy functions

Type of copy	**Copy function names**
Dynamic copy of data	XRC, CXRC, PPRC, Global Mirror
Point-in-time copy of data	FlashCopy, SnapShot, and Concurrent Copy

Note: Advanced Copy Services largely relies on a data movement engine, the system data mover (SDM), to efficiently and reliably move large amounts of data between storage devices. For more information, see "System data mover" on page 28.

Extended remote copy (XRC)

Remote copy offers two options for your disaster recovery and workload migration needs: extended remote copy (XRC) and peer-to-peer remote copy (PPRC). XRC addresses the problem of unrecoverable data that occurs between the last, safe backup of a primary system to a recovery system and the time when the primary system fails.

XRC provides an asynchronous copy operation, over distance, with minimal performance impact to primary system DASD I/O write operations.

XRC supports only System z-attached (count key data, or *CKD*) devices. This includes data running on z/OS, z/VM® and Linux.

XRC functions can be dispatched on a zIIP processor if one is available. To enable XRC for operating on a zIIP processor, use the zIIPEnable parameter in the ANTXIN00 PARMLIB member, as shown in Table 22 on page 127. For more general information about zIIP processing, refer to the section about using the System z Integrated Information Processor in *z/OS MVS Planning: Workload Management*. See also *z/OS RMF Report Analysis* for information on how to monitor zIIP processing.

For additional information about:

- XRC, refer to Chapter 2, "What is remote copy?," on page 11.
- Planning for XRC, refer to Chapter 3, "Planning for extended remote copy," on page 25.
- Setting up your XRC environment, refer to Chapter 4, "Setting up the extended remote copy environment," on page 49.
- Issuing XRC commands, refer to Chapter 5, "Extended remote copy command descriptions," on page 69.
- Managing your XRC environment, refer to Chapter 7, "Managing extended remote copy operations," on page 157.
- Managing Coupled XRC sessions, refer to Chapter 8, "Managing coupled extended remote copy sessions," on page 207.

- Recovering your data with XRC, refer to Chapter 9, "Extended remote copy data recovery operations," on page 241.
- Migrating data with XRC, refer to Chapter 10, "Migrating data with extended remote copy," on page 251.
- Recovering from error conditions using XRC, refer to Chapter 11, "Recovering from error conditions using extended remote copy," on page 255.

Peer-to-Peer Remote Copy (PPRC)

Like XRC, Peer-to-Peer Remote Copy (PPRC) addresses the problem of unrecoverable data that occurs between the last, safe backup of a primary system to a recovery system and the time when the primary system fails.

PPRC offers several options supporting copy functions both synchronous and asynchronous with the primary volume's I/O operation.

PPRC supports both System z-attached (CKD) devices and Open System (fixed block, *FB*) devices.

For additional information about:

- PPRC, refer to "Overview of remote copy" on page 12.
- Planning for PPRC, refer to Chapter 12, "Planning for Peer-to-Peer Remote Copy," on page 263.
- Setting up your PPRC environment, refer to Chapter 13, "Setting up the Peer-to-Peer Remote copy environment," on page 281.
- Issuing PPRC commands, refer to Chapter 14, "Peer-to-Peer Remote copy command descriptions," on page 291.
- Managing your PPRC environment, refer to Chapter 15, "Managing Peer-to-Peer Remote Copy operations," on page 321.
- Recovering your data with PPRC, refer to Chapter 16, "Peer-to-Peer Remote Copy data recovery operations," on page 367.

FlashCopy

The FlashCopy function provides a point-in-time copy of data for backup and recovery operations. FlashCopy creates a *Time 0* (T0) copy of a volume or set of tracks. The T0 copy is current with the volume or set of tracks at the time that the action is initiated. Subsequent updates to the original volume or set of tracks are not reflected in the T0 copy.

In contrast to FlashCopy Version 1, FlashCopy Version 2 allows you to copy individual tracks and create multiple relationships. The target tracks for a FlashCopy V2 request can be on the same volume as the source, or any other volume within the same ESS subsystem. Also, multiple source copies can be made to different target locations. Restrictions on source and target volumes being in the same logical subsystem have been lifted. Target tracks do not need to be on a different volume and have the same track addresses as the source tracks. This allows you to copy only critical data instead of an entire volume, and it allows the copies from multiple source volumes to share the same target volume.

FlashCopy supports both System z-attached (CKD) devices and Open System (FB) devices.

For additional information about FlashCopy, refer to Chapter 26, "What is FlashCopy?," on page 471.

Global Mirror

A Global Mirror environment also addresses the issue of unrecoverable data lost between backups. Global Mirror gives you the ability to establish PPRC-XD pairs over a long distance and maintain consistent FlashCopy versions at the disaster recovery site without suffering any significant performance degradation or requiring applications to be quiesced.

For additional information about Global Mirror, refer to Chapter 20, "Planning for Global Mirror," on page 407.

Metro/Global Copy

The Metro/Global Copy function allows you to *cascade* a PPRC pair with a PPRC-XD pair such that the PPRC secondary also serves as the PPRC-XD primary. In this configuration, a primary and secondary pair is established with the secondary located in a nearby site, protected from primary site disasters. The secondary volume for the PPRC-XD copy could be located thousands of miles away and would continue to be updated if the original primary location suffered a disaster.

For additional information about Metro/Global Copy, refer to "Metro/Global Copy" on page 271.

Metro/Global Mirror

The Metro/Global Mirror function enables a three-site, high availability disaster recovery solution. It combines the capabilities of both Metro Mirror and Global Mirror functions for greater protection against planned and unplanned outages.

For additional information about Metro/Global Mirror, refer to Chapter 25, "Metro/Global Mirror," on page 451.

Multi-Target Mirror

A Multi-Target Mirror environment allows a device to be the primary of more than one PPRC pair. It simplifies recovery scenarios for cascaded configurations, when swapping from the local site to the intermediate site. With Multi-Target support, during a failover to the intermediate site, the intermediate device can become a primary to both the local and the remote, eliminating the need to suspend the intermediate to remote site pair when swapping back to the local site. It provides for the capability of an incremental resynchronization between the two secondary volumes in both planned and unplanned swap scenarios.

Failover/Failback

Metro Mirror and Global Mirror for ESS support the capability to initiate *failover* to the secondary site in the case of a disaster or planned outage at the primary site. Once the primary site is recovered, you can initiate *failback* to the primary site from the secondary site, restoring your original primary-secondary volume relationships. See "PPRC failover/failback" on page 370 for details.

Although failover/failback is supported by the ESS, it is intended for use with Global Copy only in a planned outage scenario.

SnapShot

SnapShot is a function of RAMAC Virtual Array (RVA) storage subsystems that allows you to make a very quick copy of a set of tracks (an entire volume, a data set, or just a random set of tracks). The copy operation is completed with only a few I/Os to the device.

SnapShot supports only System z-attached (CKD) devices.

For additional information about:
- SnapShot, refer to Chapter 27, "What is SnapShot copy?," on page 525.
- SnapShot requirements, refer to "SnapShot copy requirements" on page 526.
- Using Snapshot, PPRC, and RVA together, refer to "Using SnapShot copy, PPRC, and RVA together" on page 529.

Concurrent copy

Concurrent copy is an extended function that enables data center operations staff to generate a copy or a dump of data while applications are updating that data. Concurrent copy delivers a copy of the data, in a consistent form, as it existed before the updates took place.

Concurrent copy supports only System z-attached (CKD) devices.

For additional information about concurrent copy, refer to Chapter 28, "What is concurrent copy?," on page 533.

Using the ESS Copy services Web interface

IBM includes a Web-browser interface called TotalStorage Enterprise Storage Server (ESS) Copy Services. The interface is part of the ESS subsystem and can be used to perform FlashCopy and PPRC functions.

The default support for the Web Copy Services is for open-systems environments. If you plan to use Copy Services functions using the TotalStorage ESS Copy Services interface, ensure that your IBM service representative enables the proper support to use System/390® volumes. The *IBM TotalStorage Enterprise Storage Server Introduction and Planning Guide* contains a worksheet for ESS Copy Services that helps the service representative determine which option to enable.

For additional information about using the Web-browser interface, refer to the *IBM TotalStorage Enterprise Storage Server Introduction and Planning Guide*.

Combining copy services operations

Copy services include Concurrent Copy, SnapShot, FlashCopy, XRC, and PPRC. Various hardware platforms support different combinations of these capabilities as shown in Table 2 on page 8. In the table, FlashCopy refers to either FlashCopy V1 or FlashCopy V2, FlashCopy V1 refers to volume FlashCopy, and FlashCopy V2 refers to data set FlashCopy as documented in Chapter 26, "What is FlashCopy?," on page 471.

Table 2 on page 8 represents:
- Copy services capabilities supported by TSO commands described in this document, and not the support for the copy services API macro ANTRQST, the program ANTTREXX and the IBM TotalStorage ESS Web-browser interface. For

information about the ANTRQST macro, refer to Appendix C, "ANTRQST and ANTRQSTL macros – call to the system data mover API," on page 573. For information about the ANTTREXX program, refer to Appendix D, "REXX support for the ANTRQST API," on page 745.

- What copy services combinations are allowed on the same device. In some cases, the indication of **NO** for a combination might be allowed. However, this combination is not recommended by IBM because undesirable results might occur. Some of these undesirable results include suspension of XRC volumes and or sessions, and the unexpected withdrawal of tracks from FlashCopy relationships.

  To determine whether a combination works, select one volume type from the top row and one from the leftmost column. The top row represents the copy service function in which a device is currently active, and the leftmost column represents the copy service function in which a device may become active.

Table 2. Allowable Combinations of Copy Operations on the Same Device

If device is → can it also become ↓?	**XRC source**	**XRC target**	**PPRC source**	**PPRC target**	**FlashCopy source**	**FlashCopy target**	**Concurrent copy source (2)**	**SnapShot copy source**	**SnapShot copy target**
XRC source	No	Yes	Yes	No	Yes	Yes	Yes	No	No
XRC target	Yes	No (2)	Yes	No	Yes	No	Yes	Yes	No
PPRC source	Yes	Yes	Yes(4)	Yes	Yes	Yes(3)	Yes	Yes	No
PPRC target	No	No	Yes	No	Yes	Yes(5)	No	Yes	No
FlashCopy source	Yes	Yes	Yes	Yes	Yes(1)	No	Yes	No	No
FlashCopy target	No	No	Yes(3)	No	No	No	No	No	No
Concurrent copy source (2)	Yes	Yes	Yes	No	Yes	Yes	Yes	No	No
SnapShot copy source	No	Yes	Yes	Yes	No	No	No	No	Yes
SnapShot copy target	No	No	No	No	No	No	No	Yes	No

Note:

1. With FlashCopy V2, a volume can contain both source and target tracks. However, a track or extent cannot be:
   - Both a source and a target
   - The target from more than one source.
2. Host software disallows this combination within a logical XRC session. There is no protection across logical XRC sessions. Unpredictable results will occur, as data is written to the same track from two different source volumes.
3. This is allowed unless the PPRC source is Global Mirror. You cannot FlashCopy to a Global Mirror source volume, unless the PPRC source is either a Global Mirror primary or a PPRC primary in a cascaded environment.
4. With appropriate microcode levels.
5. Occurs only with remote pair FlashCopy / Preserve Mirror, and is done by the microcode. It cannot be done by software.
6. An XRC target that is also used as a FlashCopy target will produce unpredictable results. However this configuration is not prevented.

For additional information about:

- Using the Web-browser interface, refer to the *IBM TotalStorage Enterprise Storage Server Introduction and Planning Guide.*
- The ANTRQST macro, refer to Appendix C, "ANTRQST and ANTRQSTL macros – call to the system data mover API," on page 573.
- The ANTTREXX program, refer to Appendix D, "REXX support for the ANTRQST API," on page 745.

Chapter 2. What is remote copy?

Remote copy is a storage-based disaster recovery, business continuance, and workload migration solution that allows you to copy data to a remote location in real time. "Remote copy" refers to these main Advanced Copy Services functions:

1. Metro Mirror (Peer-to-Peer Remote Copy, or ***PPRC***)
2. Global Copy (PPRC-Extended Distance, or ***PPRC-XD***)
3. Global Mirror, or asynchronous PPRC
4. Global Mirror for System z (Extended Remote Copy, or ***XRC***)

and these additional functions which combine elements of those listed above with other Advanced Copy functions:

1. Metro/Global Copy (***PPRC*** and ***PPRC-XD***)
2. Metro/Global Mirror (***PPRC*** and ***Global Mirror***)
3. Multi-Target Mirror (MTPPRC)
4. Metro/zGlobal Mirror (**XRC-IR**)
5. Failover (Reverse ***PPRC*** relationship during an outage)
6. Failback (Copy changed tracks following an outage)

Note: Remote copy is an extended function on most IBM storage controls, along with the appropriate levels of DFSMSdfp and z/OS. The Coupled Extended Remote Copy (CXRC) function of XRC supports very large customer configurations. The PPRC function provides both synchronous and asynchronous support.

For additional information about:

- Coupled XRC, refer to Chapter 8, "Managing coupled extended remote copy sessions," on page 207.
- PPRC, refer to "Choosing Peer-to-Peer Remote copy" on page 18 and "Managing PPRC extended distance mode" on page 353.

In this topic

This topic provides overviews on these remote copy (XRC and PPRC) solutions, and how to choose which one to use in your environment. It includes the following sections:

Overview of remote copy

For many years the perception of data protection has centered around recovery from local events such as fire or weather. This was often addressed by having backup tapes at a remote site, which would ensure recovery in a 24 to 48 hour window. Today, when businesses are often required to be operational 24x7x365, and potential disasters due to weather, power outages, fire, water, or even terrorism pose numerous threats, the importance of real time disaster recovery and business continuance have become absolutely necessary for many businesses. The remote copy functions described herein provide the framework for solutions that address this critical business requirement.

Remote copy operates with two systems: A primary system at one location and a recovery system at another location. You can locate both systems in the same building or at remote locations. The recovery system only needs to be in place for the time when a recovery is required, and can be a stand-alone system. Each system has specific DASD that processes data that you have identified as remote copy-managed. In case of a disaster at your primary location, your recovery system accesses data from the recovery system DASD.

Once established, remote copy provides the following benefits:

- Provides a real-time copy of all SMS-managed and non-SMS-managed data.
- Copies changes of your primary system data to your recovery system as you make those changes to your primary system data.
- Is application independent. You do not need separate copy functions for multiple applications or databases.
- Supports all DASD data needed for application recovery.
- Provides an improved recovery time on the recovery system, especially in the case of a disaster.

Note: To help switch applications from the primary site to the recovery location, you may want to automate operations, with a program such as NetView® or Geographically Dispersed Parallel Sysplex® (GDPS®). Your installation can make this switch with a minimal impact on users.

Using remote copy for disaster recovery

Disasters occur in many forms. Some disasters happen suddenly, when they do, they stop all processing at a single point in time. A more likely scenario happens when a disaster interrupts operations in stages that occur over several seconds or even minutes. This is often referred to as a rolling disaster. You must, therefore, plan for recovery from a potential disaster that causes system failures that are *immediate, intermittent, or gradual*. IBM's remote copy functions address this real-world situation.

Recovering from a disaster without remote copy

Figure 1 on page 13 shows how a data processing facility attempts disaster recovery without remote copy. The existing system cannot keep data at the recovery site current; therefore, any updates made since the last periodic backup may be lost. Without remote copy, you must determine what updates have occurred since the last backup, and then try to manually recreate the missing data.

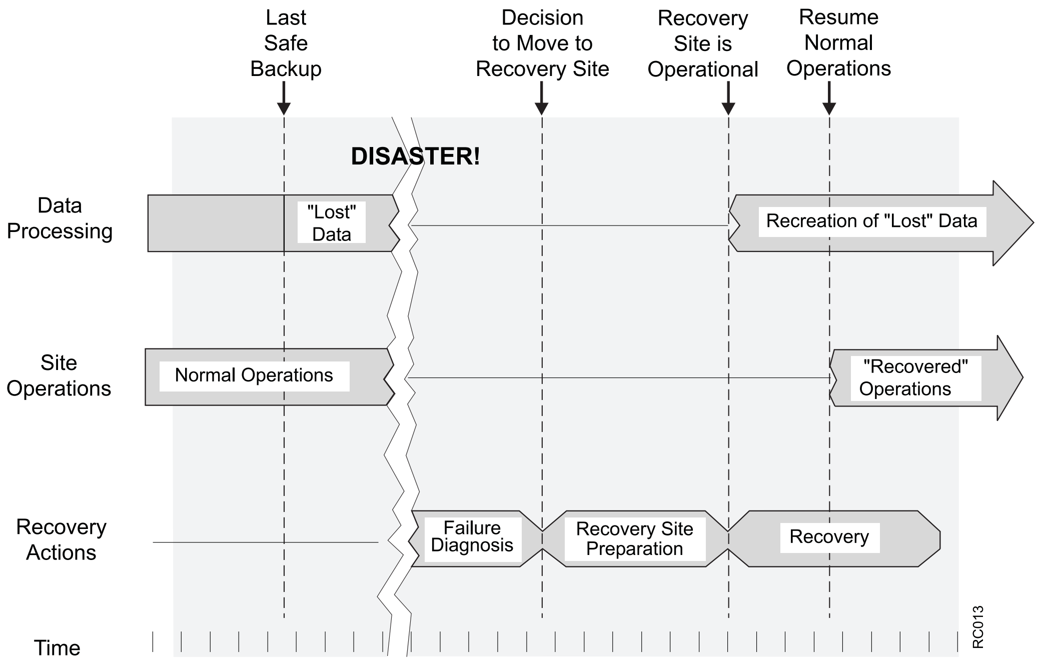


Figure 1. Current disaster recovery scenario (without remote copy). These actions portray a typical sequence of events.

Recovering with remote copy

Figure 2 on page 14 shows how disaster recovery proceeds with remote copy solutions.

Comparing Figure 1 with Figure 2 on page 14, you find that remote copy improves the current disaster recovery process by allowing a much shorter recovery time with little or no data loss. Remote copy promotes faster recovery because it constantly updates the records at the recovery site to match the primary application records. In the event of a disaster, remote copy minimizes or even eliminates data loss.

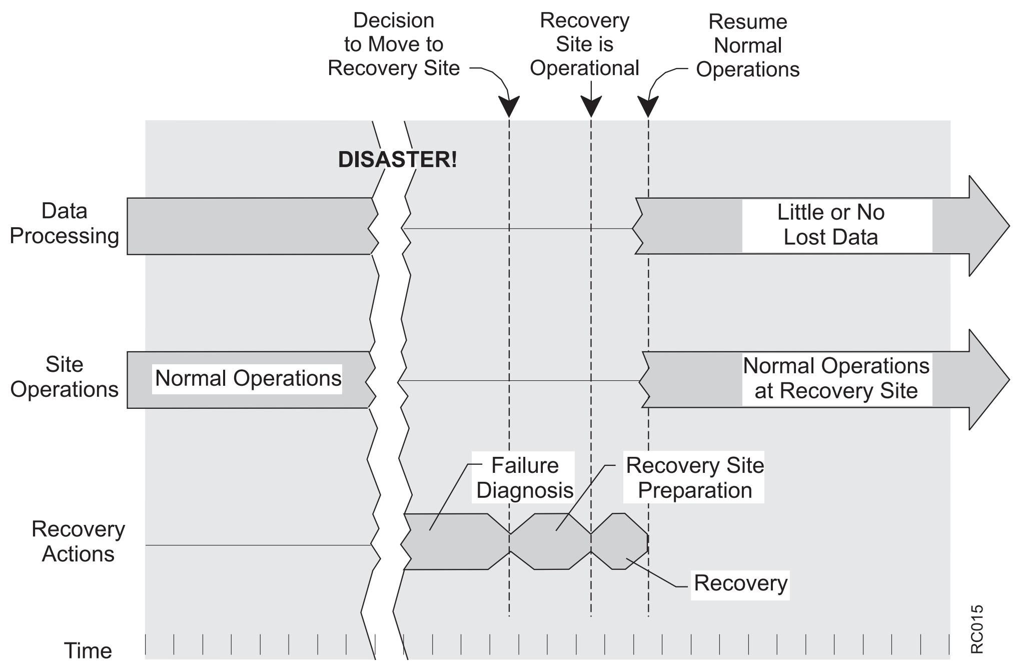


Figure 2. Disaster recovery scenario with remote copy. Remote copy allows recovery to take place faster and more completely.

The remote copy XRC and PPRC options maintain current copies of your vital data as follows:

- With synchronous PPRC, a minimal amount of data might be lost between the last update at the primary system and the recovery at the recovery system site.
- With PPRC extended distance, data updates might be lost since the last complete copy cycle; because updates are not copied in order, the contents of the secondary volume are unpredictable.
- With XRC, the only data that is lost is data that is in transit between the time that the primary system fails and the recovery at the recovery system site occurs.

For additional information about PPRC, refer to Chapter 15, "Managing Peer-to-Peer Remote Copy operations," on page 321.

Using remote copy for workload migration

In addition to disaster recovery, you can use remote copy for migrating the workload from one location to another or for moving (or migrating) data from one set of volumes to another, with minimal effect on continuous availability applications. For example, reasons for migrating data from one volume to another might include the following:

- Removing volumes from use
- Performing maintenance on a storage control
- Moving applications to a new data center in another location

For additional information about:

- Migrating a remote-copy DASD volume to any other DASD type, refer to Chapter 10, "Migrating data with extended remote copy," on page 251.
- Migrating work from the primary site to a secondary site, refer to Chapter 17, "Moving and migrating data with Peer-to-Peer Remote Copy," on page 373.

- Redirecting application I/O from one set of DASD volumes to another set of the same type, refer to Chapter 19, “Peer-to-Peer Remote Copy dynamic address switching (P/DAS),” on page 389.

Restrictions for migrating data

Remote copy can copy data from one volume to another if the number of bytes per track and the number of tracks per cylinder are the same between the two devices, and the target device has an equal or greater number of cylinders. The only way to copy data between unlike devices or from larger devices to smaller capacity devices is by using DFSMSdss (or DFDSS, or a comparable product) logical dump, restore, or copy functions. Although the 3380 volumes do not have the same number of bytes per track as the 3390 volumes, you can use DFSMSdss to copy data between the two unlike devices.

For additional information about DFSMSdss copy functions, refer to the *z/OS DFSMSdss Storage Administration*.

Advantages of using remote copy to migrate data

Using remote copy over other data migration methods provide the following advantages:

- Minimal application down time — Remote copy updates the remote migration copy at the same time that you update the application. When the migration copy is complete, you simply quiesce application programs, ensure that all updates are on the migration volumes, and start up the application using the new volumes.
- Minimal application impact — Application performance is not affected during the migration. You can continue to update data normally, with a pause to switch to the new volumes.
- Migration automation — There is less operator intervention that is required to perform the migration because there is no need for a dump and restore process.

Deciding which remote copy option to use

Based on your specific needs, you can choose from several remote copy disaster recovery options: Extended remote copy (XRC), peer-to-peer remote copy (PPRC), or PPRC-Extended Distance.

Both primary and recover site storage control must be on cached storage subsystems.

Table 3 compares both the extended remote copy and peer-to-peer remote copy options.

Table 3. Comparison of Remote Copy Functions

Topic . . .	XRC	PPRC	PPRC-XD	Global Mirror	Cascaded PPRC	Multi-Target PPRC
Application independent?	Yes	Yes	Yes	Yes	Yes	Yes
Type of solution	HW and SW	HW and SW	HW and SW	HW and SW	HW and SW	HW and SW
Design priority	Minimize impact on application performance	Maintain current data at recovery site	Minimize impact on application performance	Minimize impact on application performance	Cascade	Multi-Target
Recovery	Entire session	Vol-by-vol basis	Vol-by-vol basis	Vol-by-vol basis for entire session	Vol-by-vol basis for entire session	Vol-by-vol basis for entire session

Table 3. Comparison of Remote Copy Functions (continued)

Topic . . .	XRC	PPRC	PPRC-XD	Global Mirror	Cascaded PPRC	Multi-Target PPRC
Recovery site data is current	Except the bytes in transit	Secondary site kept fully current	Except the bytes in transit	As of last FlashCopy consistency group completed (6)	Depends on the technology used (6).	Depends on the technology used (6).
Support software for system with system data mover	All supported z/OS releases	All supported z/OS releases	All supported z/OS releases	All supported z/OS releases	All supported z/OS releases	All supported z/OS releases
Primary system support software	Must support timestamping function	All supported z/OS releases	All supported z/OS releases	All supported z/OS releases	All supported z/OS releases	All supported z/OS releases
Channel type	ESCON, fiber, or parallel	ESCON and FCP for PPRC link (ESCON, fiber, or parallel for host links)	ESCON and FCP for PPRC link (ESCON, fiber, or parallel for host links)	FCP for PPRC link (FCP for host links)	FCP for PPRC link (ESCON, fiber, or parallel for host links)	FCP for PPRC link (ESCON, fiber, or parallel for host links)
Supported copy distance	Any distance supported (2)	Recommended < 100km for performance reasons	Unlimited (3)	Unlimited	Unlimited (7)	Unlimited (7)
Copy operation	Asynchronous	Synchronous	Asynchronous (4)	Asynchronous	Asynchronous (5); for Metro/Global Mirror, synchronous for the secondary site.	Asynchronous (5); for Metro/Global Mirror, synchronous for the secondary site.
Multi-vendor support	Yes	No	No	No	No	No

Notes:

1. When PPRC extended distance is enabled, data at the secondary site is not consistent with the primary site. For more information, see "Managing PPRC extended distance mode" on page 353.
2. Adding channel extenders can extend the distance by sending the data across telecommunication lines. See "Using channel extenders" on page 20 for more information.
3. For PPRC extended distance, the distance between storage controls can be greater than that supported with ESCON. For information, see "Managing PPRC extended distance mode" on page 353.
4. The initial volume copy to the secondary device is asynchronous. Primary updates are transmitted asynchronously to secondary volumes when PPRC extended distance is enabled. For more information, see "Managing PPRC extended distance mode" on page 353.
5. For Metro/Global Copy, the initial volume copy to the local PPRC secondary is synchronous. Updates are transmitted asynchronously to the PPRC-XD secondary volumes. For more information, see "Metro/Global Copy" on page 271.
6. The secondary site kept fully current for both Metro/Global Copy and Metro/Global Mirror. For Metro/Global Mirror, the remote site is current up to the last FlashCopy consistency group created. The Global Mirror master will form FlashCopy Consistency Groups at a user-specified interval. The data at the recovery site is consistent to the last successful consistency group formed by the master. With Incremental Resync, the remote site can be made current in the case of an intermediate site outage.
7. For the Metro Mirror relationship, the distance should be < 100 km for performance reasons, and the Global Mirror relationship may be at unlimited distance.

Choosing extended remote copy

Extended remote copy is a combined hardware and software solution to the problem of accurate and rapid disaster recovery. XRC also provides a DASD and workload migration solution.

XRC is designed for sites that match the following criteria:

- Must maintain the highest levels of performance on their primary system.

- Support extended distances between volume copies.
- Can support a recovery point objective time of a few seconds.
- Provides support for a mixed vendor environment. You are allowed to mix and match primary and secondary volumes for any vendor supporting XRC architecture. Primary volumes must have XRC architecture support, but secondary volumes need not have XRC architecture support.

Protecting your enterprise's data requires that related updates are applied to the secondary volumes in the same order as they were applied on the primary volumes. Maintaining data integrity becomes especially critical when a volume is updated by multiple applications, or when a data set exists on multiple volumes spread across multiple storage controls. XRC's design strategy ensures that secondary updates are applied on a consistent basis across multiple storage controls. This update sequencing is necessary in order to avoid data integrity problems and potential data loss.

XRC supports channel extenders, ESCON, FICON®, and parallel channel operation. With ESCON and FICON channels, the XRC function provides for the recovery system to reside at an extended distance from the primary system.

Figure 3 presents an overview of the XRC option of remote copy. The workload (or DASD) migration configuration is basically the same, except that the "recovery site" would be considered the "migration target".

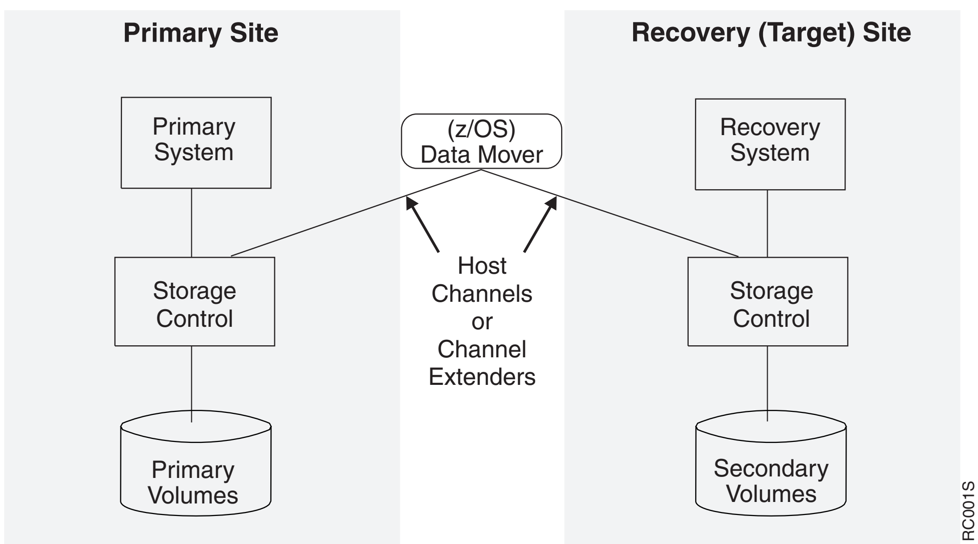


Figure 3. Extended remote copy overview

For additional information about CXRC , refer to Chapter 8, "Managing coupled extended remote copy sessions," on page 207.

How extended remote copy (XRC) works

XRC is implemented in a cooperative way between cached storage subsystems and DFSMS/MVS host system software. With XRC, copies of updated data are automatically sent to the recovery system. This is done asynchronously to data updates on the primary system, and with a minimal increase to DASD write response time at the application.

Choosing Peer-to-Peer Remote copy

PPRC, available on all supported z/OS releases, is a hardware solution to the problem of accurate and rapid disaster recovery, and also provides a workload migration solution. It is designed to provide data transfer synchronously and asynchronously.

Synchronous PPRC is designed to provide real-time mirroring of logical volumes within an ESS or between two ESSs. The sites using **synchronous** PPRC must conform to the following conditions:

- Need the recovery system to always be fully current with the primary system
- Can accept some performance impact to application write I/O operations at the primary location
- Can accept typical write overhead plus 1ms per km impact.

PPRC **extended distance** is designed for those sites that conform to the following conditions:

- Need a disaster recovery solution with a recovery point object (RPO) of many hours, or even several days.
- Have the primary and recovery storage control sites separated by very long distances beyond those supported for synchronous PPRC.
- Need the flexibility to use both synchronous and asynchronous data transfers, especially when bandwidth restrictions are a consideration.

Figure 4 presents an overview of the PPRC option of remote copy, based on a disaster recovery need. The workload (or DASD) migration configuration is basically the same, except that the "recovery site" would be considered the "migration target."

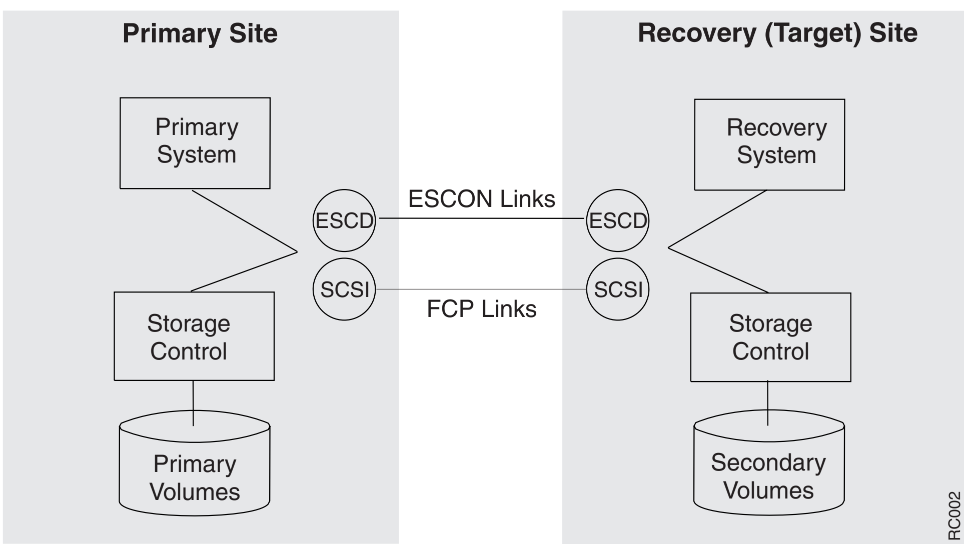


Figure 4. PPRC overview

How Synchronous PPRC works

The following describes the PPRC synchronous copy functions:

- A PPRC data copy to the recovery storage subsystem is synchronous with the primary volume's I/O operation. This means that the primary system writes

data to a primary volume's storage subsystem, the data is transferred to cache and nonvolatile storage (NVS), and the storage control sends channel end status to the host.
- The primary site storage control then initiates an I/O channel program to the recovery site storage control to transfer the updated data.
- The primary site storage control returns device end status to the primary system when the transfer to the recovery site storage control cache and NVS is complete.
- The primary system notifies the application program that the operation is complete.

Note: The PPRC copy function does not consider the primary system DASD write operation complete until the data that is sent to the recovery storage subsystem has received channel end and device end status from the secondary volume storage control. Each primary system write to the recovery subsystem causes an increase to the primary system response time.

How PPRC extended distance works

When PPRC extended distance is active, the ESS captures information about updates to the primary and periodically sends those updates to the secondary. As a result, there is no guarantee that application dependent writes are transferred in the same sequence as they were applied to the primary volume.

For additional information about PPRC extended distance, refer to "Understanding how PPRC extended distance works" on page 354.

How Metro/Global Copy works

Metro/Global Copy is a cascaded PPRC solution that enables you to combine two PPRC pairs, synchronous PPRC and PPRC-XD, which operates asynchronously, to create a low-cost, long-distance, tier four data loss disaster recovery configuration. In a cascading PPRC configuration, a single volume is defined as both a primary and a secondary, combining three volumes into two PPRC pairs.

For additional information about Metro/Global Copy, refer to "Metro/Global Copy" on page 271.

Choosing Global Mirror

A Global Mirror environment also addresses the issue of unrecoverable data lost between backups. Global Mirror gives you the ability to establish PPRC-XD pairs over a long distance and maintain consistent FlashCopy versions at the disaster recovery site without suffering any significant performance degradation or requiring applications to be quiesced.

How Global Mirror works

Global Mirror for ESS combines PPRC-Extended Distance pairs (PPRC-XD) with FlashCopy consistency groups to provide true backup at a secondary site. This is accomplished through the addition of a master session in the hardware configuration. This master session controls all updates to the secondaries of the PPRC-XD pairs and, at the secondary site, the creation of consistent copies (using FlashCopy), at user specified intervals. Global Mirror for ESS requires FCP connection.

For additional information about Global Mirror, refer to Chapter 20, "Planning for Global Mirror," on page 407.

How Metro/Global Mirror works

Metro/Global Mirror is a three-site solution, which uses synchronous replication to mirror data between a local site and an intermediate site, and asynchronous replication to mirror data from an intermediate site to a remote site. In this configuration, a Metro Mirror pair is established between two nearby sites (local and intermediate) to protect from local site disasters. The Global Mirror volumes can be located thousands of miles away and would continue to be updated if the original local site has suffered a disaster and I/O has to be failed over to the intermediate site. In the case of a local-site-only disaster, Metro/Global Mirror can provide a zero-data-loss recovery at the remote site as well as at the intermediate site.

For additional information about Metro/Global Mirror, refer to Chapter 25, “Metro/Global Mirror,” on page 451.

Choosing Multi-Target Mirror

A Multi-Target Mirror environment allows a device to be the primary of more than one PPRC pair. It simplifies recovery scenarios for cascaded configurations, when swapping from the local site to the intermediate site. With Multi-Target support, during a failover to the intermediate site, the intermediate device can become a primary to both the local and the remote, eliminating the need to suspend the intermediate to remote site pair when swapping back to the local site. It provides for the capability of an incremental resynchronization between the two secondary volumes in both planned and unplanned swap scenarios.

Multi-Target Mirror allows for

- Synchronous replication within a data center combined with another metro distance synchronous relationship
- Adding another synchronous replication for migration without interrupting an existing replication
- Combining synchronous mirroring at metro distances with asynchronous mirroring at global distances from the same primary
- Improved support for three and four-site configurations

For more information, refer to “Multi-target PPRC solutions” on page 271.

Using channel extenders

Remote copy supports channel extenders for instances where a direct ESCON or Fiber connection is not possible, or where the performance overhead is too severe. Channel extenders perform the following functions:

- Allow you to configure remote copy between locations that are either beyond the maximum ESCON channel distance, or where ESCON fiber optic cables cannot physically be placed.
- Operate as a pair of devices. One device attaches to a host system channel and the second unit connects to the storage control, which considers the attached extender to be a system channel. Both devices are connected through a network of fiber-optic cables or telecommunication lines.

An advantage of XRC is that it minimizes the effect on applications when data moves across long distances. Channel extenders provide an important part of this solution. There may be a small delay between the time that the application makes the update and when the system data mover receives the update. This potential delay is a result of the time that is required for signals to travel greater distances and the channel-to-network protocol translation necessary to achieve transmission at the extended distance.

For PPRC extended distance operations, channel extenders are used when the primary and secondary storage control sites are separated by very long distances. For information, see “Managing PPRC extended distance mode” on page 353.

XRC, PPRC and GDPS combination incremental resynchronization

XRC, PPRC and GDPS combination incremental resynchronization (IR) offers both high availability and disaster recovery capability. It consists of a three-site solution that combines XRC and PPRC in a GDPS environment, where two sites are geographically close and a third site is geographically remote. The two close sites are connected through PPRC. The remote site is mirrored by XRC. High availability is provided by HyperSwap® function between the two close sites. Disaster recovery capability is provided for the remote site. For more information, see the licensed publication *GDPS Metro/zGlobal Mirror: Planning and Implementation Guide*, ZG24-1757.

Metro/zGlobal Mirror (XRC-IR) is not supported with multi-target PPRC on the primary.

Part 2. Extended Remote Copy

This provides the information to help you use extended remote copy.

Chapter 3. Planning for extended remote copy

Before installing and using XRC, you need to plan for it. Understanding the system requirements helps you prepare for a successful installation of XRC.

For additional information about a coupled XRC environment, refer to Chapter 8, "Managing coupled extended remote copy sessions," on page 207.

In this topic

This topic is intended to help you plan for extended remote copy. It includes the following sections:

XRC requirements

You must meet the software and hardware requirements below in order to plan for and successfully install XRC. Table 3 on page 15 includes information about XRC hardware and software requirements.

XRC software requirements

The system data mover (SDM) function works with any supported OS/390® and z/OS releases.

XRC hardware requirements

The following topics describe hardware requirements for XRC:

- ESA/390 hardware.
- Your ES/9000 environment must be compatible. The XRC primary system builds on the existing sysplex concept to ensure sequence consistency. You must have a sysplex timer if multiple processing units update XRC primary volumes. The system clock is sufficient in a single processor or virtual server environment.
- The system data mover must have access to the control, state, and journal data sets, and to the primary and secondary copy volumes. If you are going to configure a coupled environment, all SDM hosts must have access to a shared volume that contains the master data set.

- All host primary systems must have a common time reference. This is necessary so that XRC can provide data consistency for all volume updates across all attached host systems. Host system MVS/DFP software performs timestamping of all application I/O write operations to active XRC volumes. Therefore, all channel programs issued to XRC-managed volumes are timestamped with a common time reference.

  Examples: The following are examples of common time references:

  – The system time-of-day clock provides the common time reference for environments with either a single processor, or with multiple virtual servers defined.

  – In an environment with multiple processing units, the sysplex timer, or equivalent, provides the common time reference for application programs. XRC and application programs both require a common time reference.
- Although the primary application systems require a common timer reference so XRC can properly order dependent application I/Os, the XRC system itself does not need to be attached a sysplex timer.
- A compatible secondary volume must be available for each primary volume you wish to copy. The secondary volume must have the identical track capacity and number of tracks per cylinder and either the same or larger volume capacity.

XRC supported devices

For the primary subsystem, XRC supports all storage controls that have XRC-capable licensed internal code (LIC).

Recommendation: The secondary subsystem should be equivalent to the primary system storage control, with similar cache and NVS sizes, for the best XRC system performance and recoverability.

Note:

XRC operational considerations

This section addresses physical and operational limitations for XRC, including limits on the number of active volumes and sessions, and volume format restrictions.

The following limitations apply to XRC operations:

- Each volume may be an XRC primary in only one session. Each secondary volume can belong to only one XRC volume pair. However, a secondary volume in one XRC session may be a primary volume in another session. A volume cannot be both primary and secondary within the same session.
- Each logical storage subsystem (LSS) in an Enterprise Storage Server (ESS) storage subsystem can support up to 64 XRC sessions. This session limit is a combination of XRC and concurrent copy sessions. The 3990 Storage Control supports up to four XRC sessions per SSID. Each connected MVS system can manage a unique set of volumes behind the storage subsystem.
- You can define a maximum of 256 volumes for each LSS in an ESS storage subsystem. The 3990 Storage Control supports up to 128 volumes.
- Each XRC session can support a maximum of 80 storage control sessions. However, the practical limit, for a session with a full complement of buffers, is 40 storage control sessions. This limit is further reduced for storage control sessions that continuously handle very high write rates or MB/sec rates, such as those associated with data base logs, or LOGPLUS volumes (which receive

system logger streams). These kinds of storage control sessions should be balanced in a ratio of not more than 1:3 with lower activity storage control sessions within an XRC session.

- All volumes that are part of XRC copy operations must conform to the following **volume format**, **track**, and **access method** restrictions, to ensure data integrity:
  - Volumes must have a standard format for record zero (R0). Volumes with R0 data lengths longer than eight bytes can cause a track format error to remain undetected when the storage control formats the track in cache.
  - You cannot assign alternate tracks within the user area. User data can be overlaid if you assign a user track on the primary address as an alternate track for a secondary address.
  - All storage control Define Extent commands must specify normal access authorization mode. Data written to an XRC primary device while the storage control is in diagnostic or device support mode is *not* copied to the XRC secondary volume. It is therefore important to remove volumes from the XRC session before running a utility program like ICKDSF.
- XADDPAIRed primary volumes can remain offline when you issue the XSTART command for restart or the XADDPAIR command for suspended pairs. Utility volumes and secondary volumes must be online when you issue the XADDPAIR command. The system applications must have a common time reference.
- Ensure that only the system data mover can update XRC secondary volumes. Do not allow non-XRC applications to update XRC secondary volumes. Non-XRC applications may read from secondary volumes, but to avoid potential data integrity exposures, you must prevent operations from writing to XRC secondary volumes.
- Assign the secondary volume a different volume serial number than the primary system volume. This will allow both volumes to be online to the system that contains the system data mover. The XRECOVER command changes the secondary volume serial number to that of the primary system volume as part of the recovery operation. If you plan to read from secondary volumes during normal copy operations, either use a different catalog or refer to the data sets by explicit volume serial numbers.
- The DFSMS address space must already be active by the time the XRC address space becomes active. To obtain accurate System Management Facilities (SMF) accounting, the DFSMS address space must remain active for the duration of the XRC session, as XRC and DFSMS both write SMF records. The system data mover writes SMF type 42 subtype 11 records.
- You may need to build a channel command word (CCW) chain to correct the lack of timestamping in stand-alone programs and in operating systems that do not support timestamping (bypass start I/O drivers).
- **For VM Systems:** XRC can be used to mirror VM application system volumes defined either as unsupported or mini disks. The data mover can run either on a separate MVS LPAR, or on an MVS guest running on the same or different VM system. If an MVS guest system is used, the volumes must be defined as unsupported to the host VM system.

  If applications run on a native VM system, writes will not be timestamped. See "Understanding the importance of timestamped writes" on page 245 for more information. MVS and Linux operating systems do timestamp writes, even when run as guests under VM

For additional information about secondary volumes, refer to "Establishing XRC secondary volumes" on page 33. For additional information about SMF type 42 records, refer to "XRC information in SMF type 42 records" on page 567.

Introducing the XRC components

This section describes the following major system components in the XRC environment:

- "XRC primary systems"
- "System data mover"
- "XRC volume terminology" on page 30
- "XRC data sets" on page 32
- "XRC storage control sessions" on page 32

XRC primary systems

Primary systems are the host systems where an application runs. XRC can run on the primary system or on any other system that is connected to the XRC-managed primary volumes that the application is updating. If there are multiple primary systems, they must have a common clock reference (a system clock) to maintain data consistency across multiple storage subsystems.

System data mover

The system data mover (SDM) is a DFSMS/MVS component that interacts with data storage subsystems and with various advanced copy services functions to efficiently move large amounts of data. As updates occur to primary volumes, the SDM manages the process of copying those updates to secondary volumes.

The SDM ensures that updates to secondary volumes are made in the same order in which they were made to the primary volumes, maintaining sequence consistency.

The following topics describe XRC-specific SDM functions, the SDM address spaces, and the XRC sessions descriptions.

Functions that are performed by the system data mover

Table 4 describes the XRC operations and the functions that are performed by the system data mover.

Table 4. Functions that are performed by the system data mover

During this operation . . .	The system data mover . . .
Data shadowing	• Monitors all updates made to the XRC primary volumes. • Transfers the updated records to the journal data sets (only when SESSIONTYPE is XRC). • Maintains the XRC session status on the state data set. • Applies the updates to the XRC-managed secondary volumes while recording the activity in the control data set.
Disaster recovery	• Collects information about XRC session status and unapplied updates from the control, state, and journal data sets, and from the master data set, if the session is coupled. • Applies any remaining consistency groups to the secondary volumes. • Relabels the secondary volumes, and makes them ready for use at the recovery site.

Table 4. Functions that are performed by the system data mover (continued)

During this operation . . .	The system data mover . . .
Migration	• Monitors all updates that are made to the XRC primary volumes. • Maintains record status on the state data set. • Applies the updates to the XRC-managed secondary volumes.
Migration recovery	Relabels the secondary volumes and makes them ready for use at the recovery site.

Address Spaces for XRC

The system data mover for XRC operates in specific system address spaces that start with “ANT.” The functions that are provided by these address spaces are described in Table 5.

Note: The system data mover generates messages that begin with “ANT.” *z/OS MVS System Messages, Vol 1 (ABA-AOM)* describes functions that the address spaces provide.

Table 5. Address space descriptions

Address space	Description
ANTAS000	• This address space handles TSO commands and API requests that control XRC. • ANTAS000 and the point-in-time address space, ANTMAIN, are automatically started during IPL. • The SDM automatically reinitializes the address space if it is cancelled.
ANTCL*nnn*	• This address space manages XRC sessions within a logical partition (LPAR mode), which are to be coupled to an XRC master session as a single cluster session. • ANTCL*nnn* is automatically started at IPL time and when ANTAS000 is cancelled if the PARMLIB value of ClusterMSession indicates something other than DISABLE.
ANTAS*nnn* refers to: ANTAS001 through ANTAS020	An ANTAS*nnn* address space will start for each XSTART command issued.

Note:

1. Because they process a large number of I/O requests, ANTAS000 and ANTAS*nnn* (ANTAS001 through ANTAS020) are non-swappable. The ANTAS*nnn* address space runs at the MASTER dispatching priority. ANTAS000 is subject to normal SRM priority control.
2. You can run up to 20 XRC sessions with address spaces of ANTAS001 through ANTAS020 in a single logical partition, with or without coupling them.
   - If you enable the clustering, you can run up to 19 XRC sessions.
   - If you disable the clustering, you can run up to 20 XRC sessions.

   Other factors such as real storage capacity, auxiliary storage capacity, and available processor capacity may limit the effective number of XRC sessions to fewer than the architectural limits noted in this section.

3. If the ANTAS000 address space is canceled, XRC automatically restarts it. Except for TSO commands and API requests, there is no impact to SDM functions that run in the ANTAS*nnn* address space. If an event cancels the ANTAS*nnn* address space, XRC ends all active storage control sessions with storage controls that do not support hardware bit mapping. For hardware subsystems that support hardware bit mapping, the system replaces the volumes in a suspended state with change recording active in the subsystem.
4. If an ANTCL*nnn* address space is canceled, the system suspends all volumes for any ANTAS*nnn* address space that is coupled to a master session through the cluster session.

XRC session state descriptions

An XRC session may be in one of several states of operation. Table 6 describes the different XRC session states.

Table 6. XRC session state descriptions

State of session	Description	For information, see...
Active	The system data mover has been initiated with an XSTART command. XRC accepts the XADDPAIR, XCOUPLE, XDELPAIR, XQUERY, XSUSPEND, XSET, and XEND commands.	Chapter 5, "Extended remote copy command descriptions," on page 69
Inactive	The system data mover has been stopped by an XEND command. The only commands accepted by XRC are the XSTART and XRECOVER commands.	Chapter 5, "Extended remote copy command descriptions," on page 69
Recovery	An XRECOVER command has initiated the system data mover, which is available to update secondary volumes with unapplied journal records. XRC only accepts the XQUERY command after you have issued the XRECOVER command. The XRC session returns to the inactive mode when the recovery function is complete.	"XRECOVER–Recovering data on the recovery system" on page 100
Suspended	The XSUSPEND(*session_id*) command with the TIMEOUT option has suspended the XRC session. The ANTAS*nnn* address space is no longer active. You can issue an XSTART command to restart the suspended XRC session or an XRECOVER command to recover the session at a recovery site.	"XSUSPEND–Suspending volumes or sessions" on page 114

XRC volume terminology

The following terms occur throughout this publication to describe volumes and volume pairs that belong to an XRC session.

XRC volume descriptions

Table 7 describes the volume types.

Table 7. XRC volume descriptions

Volume type . . .	Description . . .
Primary	The source volume that XRC copies to a remote location. This is the volume to which application programs direct their updates.

Table 7. XRC volume descriptions (continued)

Volume type . . .	Description . . .
Secondary	The target volume that receives the updates that are made to the primary volume. See "Establishing XRC secondary volumes" on page 33 for more information.
Source	Another name for the primary volume.
Target	Another name for the secondary volume.
Active	A primary volume in an XRC session that receives data that is also copied to a remote location. The status of an active volume can be DUPLEX, SEQCHECK, COPY, or PENDING.
Suspended	An established primary volume in an XRC session whose data XRC is no longer copying to a remote location. The volume remains in the XRC session. See "XSUSPEND–Suspending volumes or sessions" on page 114 for more information.
Utility	A primary XRC volume that is available to the system data mover for reading data from a storage control. For more information about utility volumes, see "Using XRC utility devices" on page 165.
XRC	An established primary and secondary volume that is identified by the XADDPAIR command. The status of a volume pair can be COPY, DUPLEX, PENDING SEQCHECK, or SUSPENDED. You have the option to copy an XRC volume pair to a remote location.

XRC volume pair status descriptions

Table 8 describes the status of a volume.

Table 8. XRC volume pair status descriptions

Volume pair status . . .	Description . . .
COPY	A volume pair that XRC is currently synchronizing. Volume pairs in COPY status are not part of the XRC session and, are therefore, not eligible for recovery. XRC does not consider copy volume pairs during ERRORLEVEL processing, nor does it include them during XRECOVER and XADVANCE processing.
DUPLEX	A volume pair for which volume synchronization or resynchronization has completed and that has been successfully recovered, or that was established with the NOCOPY option. The pair is active, and is neither SUSPENDED, PENDING, nor in COPY status. The volume pair is consistent up to the session-specified timestamp reported in message ANTQ8231I.
PENDING	A volume pair that XRC has not yet synchronized or resynchronized. Volume pairs in PENDING status are not part of the XRC session and, are therefore, not eligible for recovery. XRC does not consider a volume pair in PENDING status during ERRORLEVEL processing, nor does it include the volume pair during XRECOVER and XADVANCE command processing.
SEQCHECK	A volume pair that contains nontimestamped writes. This temporary condition indicates that data consistency is uncertain, for this volume pair, in the event of a disaster recovery. The pair automatically returns to the DUPLEX status when XRC successfully writes a timestamped update to the secondary volume.

Table 8. XRC volume pair status descriptions (continued)

Volume pair status . . .	Description . . .
SUSPENDED	An established volume pair in an XRC session where XRC is no longer copying data from the primary volume to the secondary volume. The volume pair remains in the XRC session.

XRC data sets

This section defines terms for the XRC data sets. An XRC system requires one of each of the state, control, and journal data sets. A coupled XRC system requires a single master data set in addition to each of the state, control, and journal data sets per XRC subsystem. If a cluster session is to be used to couple XRC sessions in a logical partition, you require a cluster control data set. You can also optionally specify a cluster monitor data set.

XRC data set descriptions

The following terms apply to the data sets in an XRC session.

Data set type	Description
State data set	The state data set contains status of the XRC session and of associated volumes that XRC is managing. The state data set is updated if an XADDPAIR, XDELPAIR, XSET, XSUSPEND, XRECOVER, or XEND command is issued, or whenever a volume state changes. If a session is coupled, the state data set identifies the master session it is coupled to, and is updated when an XCOUPLE command is issued.
Control data set	The control data set contains consistent group information on the secondary volumes and the journal data set. It contains information necessary for recovery operations. The control data set acts as the table of contents for the session. **Example:** The control data set keeps track of data written to secondary volumes, the location of unwritten data in the journal set, and which group to start recovery with.
Journal data set	The journal data set contains temporary user data that is created when records in the primary volume are changed or updated.
Master data set	The master data set ensures recoverable consistency among all XRC subsystems contained within the Coupled XRC system.
Cluster data set	The cluster data set is used to restart a cluster session.
Cluster state data set	The cluster state data set contains XRC monitor data for a cluster session.

XRC storage control sessions

This section defines terms for XRC storage control sessions that are associated with primary volumes in an XRC volume pair. A storage control session is a logical entity that is created for processing updates to the XRC primary volumes. It is used to group sets of primary XRC volumes that are being processed by an XRC session within the storage control.

The number of storage control sessions that may be in effect for a single storage control depends on the capability of the storage control. For example, IBM 3990 and 9390 Storage Controls can each manage a maximum of four XRC storage control sessions. Each logical storage subsystem (LSS) in an Enterprise Storage Server (ESS) storage subsystem can manage a maximum of 64 storage control sessions (XRC or CC) in any combination.

Storage control session states

The following states apply to the XRC storage control sessions for a primary volume in an XRC session.

Session state . . .	Description . . .
Quiesced	A non-ESS storage control session is in quiesced state after an XSUSPEND command has completed. The session remains quiesced until either you issue an XSTART command and the storage control session becomes active, or the quiesce timeout condition expires and the storage control session ends.
Suspended storage control session	An ESS storage control session is in a suspended state after either receiving a system reset or when the XRC session becomes suspended. When XRC stops reading from the ESS, the ESS suspends the storage control session. The session remains suspended until you issue an XADDPAIR command for a volume that is associated with the storage control session.
Active	The state in which normal mirroring is done.
Terminated	The session has ended.

Establishing XRC secondary volumes

All XRC secondary volumes must stay dedicated exclusively to XRC use. It is recommended that you make the secondary volumes offline and unavailable to other host systems, but not to the system data mover, thereby ensuring that *no* write I/O operations occur outside the SDM function. You can do this by making the volumes part of a system-managed, disabled storage group, and by keeping the volumes offline to all but the SDM system. Ensure the data integrity of the secondary volumes by not allowing other functions to make updates to them.

Secondary volumes may have data read from them while they are in an XRC session. However, if primary system programs change the VTOC pointers, data can appear to have been copied improperly when read from another system. To avoid this, refresh the in-storage system pointers by varying the secondary volumes off and then on to the system that is reading from the volume. Allow the volumes to reach duplex state before doing this refresh operation.

For additional information about storage groups, refer to the *z/OS DFSMSdfp Storage Administration*.

Evaluating remote copy with DFSMShsm

If any remote copy-eligible volumes are also eligible for either primary space migration or interval migration, you must make backup copies of these migration-eligible data sets available to the disaster recovery site. The need for these backup copies can be better understood when we compare the DFSMShsm migration process to the remote copy process.

Note: Although backups are always useful, remote copy (especially XRC) reduces the need for them by keeping the offsite copy far enough away from the original to avoid the loss of both copies. The only need for backups, therefore, is to allow retrieval of older versions of existing data sets.

DFSMShsm migration from a primary volume consists of the following steps:
1. A move or delete from the primary volume
2. A change in catalog pointers to a DFSMShsm-managed control data set
3. The reallocation of the data on another DASD device or tape volume

If you are also copying the primary volume with remote copy, the DFSMShsm migration actions appear as a delete of the primary as far as remote copy is concerned. Since the secondary volume is a track-for-track image of the primary volume, the delete performed against the primary volume during the migration process is then reflected to the remote copy secondary volume, which is then deleted. You must take steps to ensure that you are not left without current copies of these data sets.

The best solution is to have policies in place that control how volumes can be removed from or added to remote copy control. One way is to take advantage of DFSMShsm management-class and storage-group SMS constructs, both of which have options to prevent automatic migration of these data sets.

Sometimes data is written to DASD for a very short length of time, as with a temporary work file or when the NOBACKUP keyword is specified. This might also be the case if the volume using remote copy services is being managed under Tape Mount Management (TMM). With TMM migrating the data every hour without backup, you must ensure that a valid backup is available to the remote recovery site.

Recommendation: An excellent way to achieve this backup is to use a fast replication copy function (such as concurrent copy, FlashCopy, or SnapShot copy) to copy the secondary volumes before migration.

Ensure that you have adequate cache capacity to accommodate both functions, as the cache resources must now serve both remote copy and concurrent copy. Without adequate policies in place, data sets available for migration or deletion by DFSMShsm cannot be guaranteed to be covered by remote copy.

Defining storage requirements

This section defines the storage requirements that are needed to support XRC.

Virtual storage requirements

XRC operates in up to 21 address spaces on a single logical partition (LPAR mode). ANTAS000 is always active and processes XRC commands. You can start from 1 to 20 XRC sessions at a time. Each unique XRC session that is started on a system occupies its own address space. (For example, the first address space is ANTAS001, the second is ANTAS002, and so forth through ANTAS020). One of these 20 address spaces is used by the ANTCL*nnn* address space if XRC clustering is enabled. See "Address Spaces for XRC" on page 29 for specific information about XRC session address spaces.

The ANTAS000 address space uses 65 MB of virtual storage, which includes a 20 MB trace data space.

Recommendation: Do not restrict the virtual storage that is used by the ANTAS*nnn* address spaces. However, if you need to limit virtual storage usage, the following provides a guideline as to the amount of virtual storage that is needed by each ANTAS*nnn* address space.

An ANTAS*nnn* address space will use approximately 1.7 GB for an XRC session. The minimum virtual storage usage will be 200 MB + (60 KB * TotalBuffers). The maximum will be up to the amount of virtual storage available on the system below the 64-bit bar. The exact amount of storage used depends on the number of storage control sessions and the amount of data being moved. If more than 1.7 GB is required, it is recommended that additional Coupled/Clustered XRC sessions be used.

Note: The numbers for the ANTAS*nnn* address space are relative to running XRC sessions with standard IBM 3390-3 size devices. Larger amounts of virtual storage are used by XRC when devices with larger numbers of cylinders are added to a session. Also, the update activity of storage control sessions may cause an ANTAS*nnn* address space to obtain virtual storage in a fragmented manner, and therefore cause XRC to use additional virtual storage.

Ensure that local paging space on the SDM is sufficient to support your planned remote copy configuration.

XRC needs a minimal amount of space (0.2 KB) in the extended common service area (ECSA). All other storage used is within the XRC address spaces. The ANTCL*nnn* address space requires a minimum of 182 KB of extended common service area (ECSA).

XRC supports a maximum of 80 storage control sessions for each XRC session.

Real storage requirements

XRC uses virtual storage buffers for data movement. At the time these buffers are used for I/O operations, they must be backed by real storage. The amount of real storage required at a given point depends on several factors, including:

- XRC PARMLIB settings
- The number of volume synchronization tasks currently active
- The number of storage control sessions in the XRC session
- The number of journal data set pairs defined for the session and the blocksize of the data sets
- The rate of data movement and the size of data blocks moved
- The duration of delays, including those associated with maintaining coupled session consistency, and those associated with writing to journals and secondary volumes

There are several types of buffers: those used to construct channel programs, those used to synchronize volumes, those used by access methods to write to the journal/state/control data sets, and those used in mainline data movement to read updates from storage control cache. This last type accounts for most of the real storage required by the system data mover.

Data movement buffers are 60K bytes. The number of buffers is controlled by PARMLIB value BuffersPerStorageControl, multiplied by the number of storage control sessions, with an overall limit imposed by the TotalBuffers PARMLIB value. A fully-configured session will have 25000 buffers, which requires approximately 1500MB of real storage to page fix. The buffers are fixed and freed as the session requires. Peak buffer demand for a session may not occur for several hours, or even days after the session is started.

To improve performance, an installation may use PARMLIB values PermanentFixedPages and ReleaseFixedPages(NO) to indicate that data movement buffers are to be kept page fixed even during periods of low activity.

Each data mover may use up to an additional 100MB of real storage above that required for data movement buffers. This makes a total real storage requirement of 1700MB per fully-configured XRC session.

Given the large real storage requirements of fully-configured XRC sessions, the potential exists for pageable storage shortages and performance degradation on systems that have not been configured with sufficient real storage. To avoid these situations, IBM makes the following recommendations:

- Use z/Architecture® systems with 64-bit real storage.
- Reserve at least 20% of total real storage for use by other operating system components.
- Reserve sufficient additional real storage for any non-XRC workload that will run on the XRC system.
- Run minimal other workload until all XRC buffers have been page fixed.

Should pageable storage shortages be observed, consider the following actions:

- Increase the real storage available to the system.
- Move non-XRC workload to a different system, or quiesce the workload until all XRC buffers have been page fixed.
- Reduce the number of XRC sessions running on a system by issuing the XSUSPEND TIMEOUT command for one or more sessions. Then restart the sessions on a different system with sufficient resources.
- Decrease the fixed storage requirement during light activity periods by reducing the PermanentFixedPages PARMLIB value. Note that this may degrade session performance during heavy activity periods.

Paging space requirements

XRC requires additional paging space to support the trace data space and the data movement address spaces. Keep in mind that there can be from two to twenty-one address spaces involved.

Table 9 shows the data and address space requirements for two address spaces.

Table 9. Address space requirements

Address space	Data space requirements	Address space requirements
ANTCL*nnn*	15 MB	1 MB
ANTAS*nnn*	15 MB	up to 2 GB per address space

Recommendation: SDM paging should use expanded storage.

For additional information about address spaces, refer to “Address Spaces for XRC” on page 29.

Library requirements

The XRC function resides in the following libraries:

- SYS1.NUCLEUS (increases by 0.1 KB)
- SYS1.LINKLIB (increases by 2 MB)
- SYS1.LPALIB (increases by 0.2 KB)
- SYS1.CMDLIB (increases by 100 KB)

Configuring extended remote copy

A critical aspect of recovery system performance (or "target" system performance, for data migration tasks) is the configuration of the full XRC system. This section describes basic XRC configurations, and those that include DASD channel extenders.

For additional information about a coupled XRC configuration, refer to "Choosing a coupled XRC configuration" on page 209.

Choosing an XRC configuration

In XRC configurations, different host systems can share common primary disk volumes. The system data mover collects all application updates and sends them to the secondary volumes.

Figure 5 on page 38 outlines three XRC configuration variations. The major difference between these configurations is the location of the system data mover.

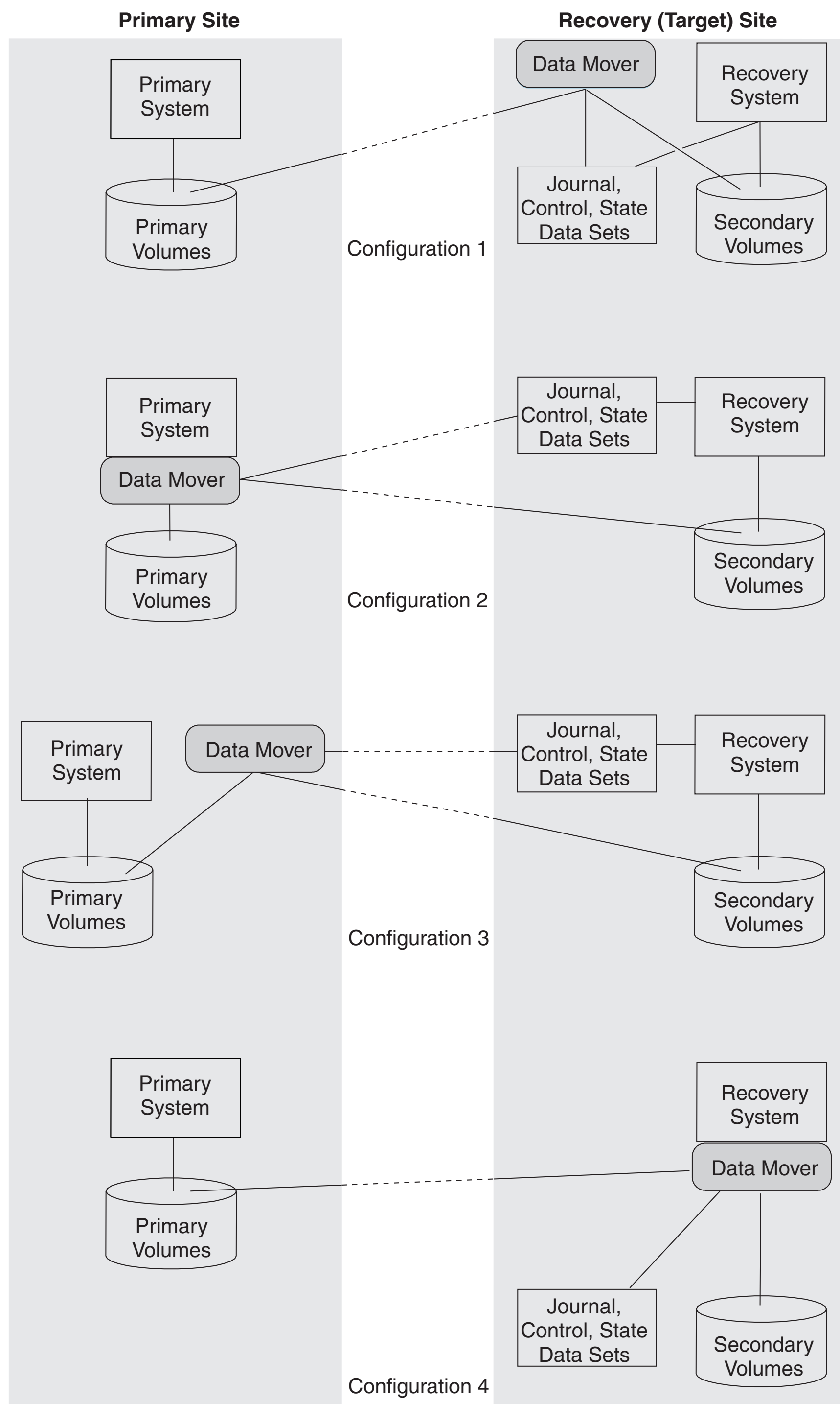


Figure 5. Basic XRC configurations

Each of the configurations in Figure 5. Basic XRC configurations places the system data mover at a different location, as explained in Table 10.

Table 10. System data mover locations in basic XRC configurations

Configuration . . .	**Location of system data mover . . .**
Configuration 1	The SDM resides at the recovery site.

Table 10. System data mover locations in basic XRC configurations (continued)

Configuration . . .	Location of system data mover . . .
Configuration 2	The SDM is on one of the application site host systems.
Configuration 3	The SDM is at the primary site, but not on the primary host system itself.
Configuration 4	The SDM is on the recovery system.

Recommendation: XADDPAIRed primary volumes can remain offline when you issue the XSTART command for restart or the XADDPAIR command for suspended pairs. Utility volumes and secondary volumes must be online when you issue the XADDPAIR command. Having the utility volumes online helps to establish the path grouping that is necessary to set up XRC sessions and, if using Enhanced Readers with aliases, keep aliases bound to the utilities' base address.

Rule: All secondary volumes must be online when the XADDPAIR command processes. You can then vary the secondary volumes offline to protect them from updates that originate from non-XRC address spaces, if needed. You can use any other non-XRC volumes on the systems at either site for local applications, as they are independent of XRC activity.

Including channel extenders

Channel extenders work in an XRC configuration when the system data mover is at either the primary site or the remote site. An XRC configuration with a channel extender operates most efficiently when the SDM software runs on a system at the secondary location. The system data mover then accesses the primary disks through the channel extender connection.

There may be situations where you cannot or do not want to have a continuously running host at the secondary site. If so, you can run the SDM on a system at the primary location with the channel extender that is connected to disk at the secondary location. This setup, however, is much less efficient than the previously mentioned configuration.

High quality channel extenders are transparent to the I/O software (except for the performance cost). XRC software can access primary or secondary disks that are configured through extenders.

For additional information about DASD channel extenders, refer to "Using channel extenders" on page 20.

Determining XRC resource needs

This section contains information and recommendations about the resources needed to establish an XRC configuration that meets disaster recovery and workload migration needs.

Before installing Extended Remote Copy, review Table 11.

Table 11. Considerations for determining XRC resource needs

Consider . . .
"Evaluating how XRC affects the primary storage subsystem" on page 40

Table 11. Considerations for determining XRC resource needs (continued)

Consider . . .
"Evaluating how XRC affects the recovery storage subsystem"
"Determining the recovery site storage capacity" on page 41
"Determining recovery system access to resources" on page 41
"Establishing storage subsystem channel connections" on page 41
"Establishing XRC ESCON connections" on page 42
"Establishing XRC system data mover connections" on page 42
"Estimating XRC system resources for system data mover operations" on page 42
"Allocating XRC journal data sets" on page 43

Evaluating how XRC affects the primary storage subsystem

About this task

Storage subsystem use can affect overall system performance, and it is therefore necessary to determine how adding XRC activity might affect the primary storage subsystem.

Perform the following steps to determine the effect of adding XRC activity to your primary storage subsystem.

Procedure

1. Make note of the write rate for the volumes that you want to copy on each primary site storage control.
2. Add this rate, as if it was all fast write, to the total storage subsystem use.
3. Decide if this impact on the storage subsystem allows applications to achieve acceptable disk I/O performance. If it does not, you may need to redistribute volumes among storage controls or add an additional primary site storage control.

Evaluating how XRC affects the recovery storage subsystem

XRC lets you allocate recovery system volumes on any number of recovery storage subsystems. The storage controls at the recovery site do not have to have a one-to-one correlation with their application storage subsystem counterparts. Project the performance impact on recovery storage subsystems to be sure that it is within acceptable limits.

For XRC, recovery site storage subsystems must be able to accommodate the XRC volume write rate. Include all other recovery storage subsystem activity in your assessment, such as access by other applications and systems to non-XRC volumes on the same storage subsystem.

With XRC, you also need to plan for the bandwidth capacity of the journal data sets. The journal data sets, which are either striped or unstriped sequential access method (SAM) data sets, must be able to accept the write rate of copied disk from the primary system.

For additional information about XRC journal data sets, refer to "Specifying XRC journal, control, and state data sets" on page 54.

Determining the recovery site storage capacity

As part of planning for XRC installation, you must consider the number and type of disk volumes, according to the steps outlined in Table 12.

Table 12. Steps for disk volume determination

Stage	Description
Determine the number of necessary storage subsystem disk volumes.	Assume that you must copy all volumes from the primary subsystem to the recovery system. You can then consider each volume individually to determine if you will need it for recovery and subsequent application operations on the recovery system. **Example:** You do not have to copy disk volumes that contain only page data sets to the recovery site storage subsystem.
Determine the number of disk volumes to be copied to the recovery site storage subsystem.	Consider which application and support data sets will be needed to start applications on the recovery subsystem at recovery time. **Example:** User catalogs are a good example of such data sets. If the recovery subsystem cannot find data volumes associated with user catalogs, jobs will not start successfully after recovery.
Consider disk volumes.	Consider what disk volumes, such as DFSMShsm migration level 1 data sets, may be needed for recovery system operations.

Determining recovery system access to resources

Rules: To meet disaster recovery needs, recovery systems must obey the following rules:

- Be able to run the applications that use the copied data.
- Be able to process commands to the secondary volumes to end the secondary copy state and to become available for primary system allocation.
- Have access to the SDM software, TSO commands, and to the state, control, and journal data sets.
- Have access to the CXRC master data set.
- Have access to the cluster control data set if sessions are coupled through a cluster session.

For additional information about host access to the master coupled data set, refer to “Allocating the CXRC master data set” on page 215.

Establishing storage subsystem channel connections

For XRC, disk channel connections include those to primary and recovery systems. Connect both primary and recovery system disk devices to the system that contains the system data mover. For some storage controls, you can use ESCON, FICON, or parallel channels between hosts and storage controls.

Tip: It may be useful when planning an initial XRC **test** installation (with a limited number of volumes) to have both the primary and secondary volumes on the same storage subsystem, thereby conserving resources.

Establishing XRC ESCON connections

You can configure IBM storage controls to support ESCON ports (connected by multimode fiber-optic cables). ESCON Directors, if your configuration includes them, can have Extended Distance Feature (XDF) ports with singlemode fiber-optic cables for longer link lengths.

You can only use 9036 ESCON Remote Channel Extenders to convert from multimode fiber links to XDF and back, and as XDF extender links. Use these or other comparable repeaters when the distance between systems is greater than 3 km and your configuration does not include ESCON Directors.

You can only dynamically switch one ESCON Director in a path to an I/O unit. The other ESCON Director, if present, provides a static connection. Refer to the appropriate configuration planning documentation for information about ESCON configurations with ESCON Directors.

Rule: Recovery site storage subsystems must have access to the host systems that will run primary applications when the primary system is out of service. A host system at the recovery site must be able to issue recovery commands as part of recovery system takeover procedures.

Establishing XRC FICON connections

XRC can be connected with FICON. You can use Dense Wavelength Division Multipliers (DWDMs) to allow connections over a single disk fiber link. ESCON and FICON connections can be used in the same configuration if desired. Be sure to consider the speed differential in order to set up a properly balanced configuration.

FICON can handle distances up to 150 km without the performance droop typical of ESCON connections. FICON also reduces host and disk subsystem protocol communications to a minimum. Additionally, you can use a FICON bridge to extend your FICON connections to longer distances.

FICON has the advantage of being able to multiplex and transfer data from many requestors simultaneously. Native FICON interfaces may be dedicated or shared, but it is important that the right number of interfaces be configured to support the workload generated by the applications and the data mover system.

Establishing XRC system data mover connections

All primary and secondary disk subsystems require connection to the system that runs the SDM software.

Note: Some storage controls allow you to use parallel channels instead of ESCON channels for the XRC disk devices that run the asynchronous SDM software. Parallel channel connections, however, may not provide the bandwidth under load or sufficient distance for disaster protection.

Estimating XRC system resources for system data mover operations

The XRC system data mover must have sufficient I/O transfer bandwidth and have access to both primary and recovery system disk devices. A good initial estimate includes enough system resources to process approximately three times the application I/O write rate to volumes that will be XRC primary volumes.

Actual resources should be less since the SDM performs much of its I/O in more efficient multiple-I/O-transfer channel programs.

Allocating XRC journal data sets

Include disk allocation for XRC journal data sets in your resource planning. When allocating journal data sets to disk volumes, consider the total, concurrent data transfer demand of all XRC writes. For very high total write rate systems, you may have to assign journal data set volumes to multiple storage controls.

Allocating a striped SAM data set also can be worthwhile. Verify that there is sufficient recovery storage control capacity when both journal data sets and secondary copy volumes are present on any single storage control.

Experience shows that a performance improvement can be realized when journal data sets are defined with stripes, with additional performance benefit for up to eight stripes.

For additional information about journal data sets, refer to “Specifying the journal data sets” on page 55 and “Maximizing access to journal data sets” on page 67.

Protecting access to XRC commands

Each installation is responsible for preventing the unauthorized use of XRC TSO commands and for controlling ICKDSF activity to XRC-managed volumes. The following are two ways to protect access to XRC commands:

- Put the commands in a library that is protected by RACF®, a component of the Security Server for z/OS.
- Define resource profiles in the RACF FACILITY class and restrict access to those profiles.

For additional information about how to place commands in a protected library, refer to “Placing TSO commands in an RACF-protected library” on page 51. For additional information about how to define resource profiles, refer to “Defining resource profiles in the RACF FACILITY class” on page 52.

XRC enhanced multiple reader function

The XRC enhanced multiple reader function provides the SDM enhancements to XRC for the function of reading updates from the storage control side file. It addresses a disparity between application write and SDM read rates by increasing the maximum number of simultaneous reads from cache from 1 to 16. The function also reduces the impact of high volume small record size updates.

Configuration recommendations

There are two methods to enable the XRC enhanced multiple reader function:

- Use the NumberReaderTasks value in the parmlib along with parallel access volume aliases for XRCUTL devices.
- Use an explicit number of XRCUTL devices for a given SSID and storage control session name (SCSN) combination.

In the following states, you can enable the function by using either of the methods:

- The session is active, with all volumes suspended.

  The all volumes suspended state has the following two sub-states:

  – The session is suspended and the address space is inactive.

- The session is active and the address space is active.
- The session is active, with non-suspended volumes.

The configuration also enables the XRC enhanced multiple reader function. You might have multiple storage control session names for a given SSID. You can use either of the following configurations when enabling the enhanced multiple reader support:

- Combine the multiple storage control sessions into an enhanced multi-reader session.
- Keep the distinct groups of volumes as primary sessions and add auxiliary sessions to each session.

To convert the primary session into the enhanced mode, issue an XADDPAIR command to the existing XRCUTL volume for the SCSN that is to become the primary storage control session ID (SCID). If you have already merged another storage control session name into the primary session, or have added an additional XRCUTL device, the primary session is converted into enhanced mode.

For the following enablement scenarios, set the PARMLIB value AllowEnhancedReader to Yes before you enable the XRC enhanced multiple reader function.

It is important when configuring an enhanced multiple reader session to choose the primary utility volume that will be used on a permanent basis. The primary utility volume is identified with a 'p' next to the SCID in the XQUERY STORAGECONTROL DETAIL report, and cannot be deleted using XDELPAIR without first deleting all other volumes (data and auxiliary utility) in the ER session group. When you create a new enhanced reader session (on an SSID that is not in the XRC session, or using a new SCSN) the first utility volume added with XADDPAIR becomes the primary utility volume. When you convert a storage control session name (SCSN) from a single reader to an enhanced reader, the first utility volume that is re-added with the XADDPAIR command becomes the primary utility. When you combine multiple SCSN's in an SSID, the first utility volume added becomes the primary and the subsequent volumes are merged into the ER group as auxiliary utilities.

Enabling the XRC enhanced multiple reader function

The following examples demonstrate how to enable the XRC enhanced multiple reader function.

Enabling the XRC enhanced readers through explicit XRCUTL devices

To use the XRC enhanced readers, issue the XADDPAIR command of another volume in storage. Enable the support through the XADDPAIR of another volume in the SSID, with XRCUTL as the secondary device by using the same SCSN value.

If explicit XRCUTL devices are used, enable the XRC enhanced multiple reader by using the following methods:

- Issue an XADDPAIR command to the existing XRCUTL device for one of the storage control sessions on the SSID. (This storage control session becomes the primary enhanced session.)
- If there is only one storage control session name, issue an XADDPAIR command to add an additional XRCUTL volume.

- If there are multiple storage control session names, issue an XADDPAIR command to use an existing XRCUTL with a different storage control session name. (This different storage control session name is the name of the first session that has been converted to the enhanced mode.)

If there is only one SCSN, refer to the following scenarios to enable the XRC enhanced readers:

The following scenarios use AA as existing SCSN value.

If the session is suspended, there is no ANTASnnn address space, and there is only one SCSN in an LSS, use the following steps:
1. Issue an XSTART command.
2. Issue an XADDPAIR command for the XRCUTL AA. XRCUTL AA is converted to an enhanced primary session.
3. Issue an XADDPAIR command for the new XRCUTL volume with the existing SCSESSION id. Repeat this step for all new XRCUTL volumes.
4. Issue an XADDPAIR SUSPENDED command to resynchronize the remaining volumes.

If the XRC session is active, and there is only one SCSN, use the following step:
1. Issue an XADDPAIR command for the new XRCUTL volume with the existing SCSESSION id. Repeat this step for all new XRCUTL volumes.

If there are multiple SCSNs, use the following scenarios to enable the XRC enhanced readers:

The following scenarios use AA and BB as the existing SCSN values.

If the session is suspended, there is no ANTASnnn in the address space, and there are multiple SCSNs in an LSS, convert a set of storage control names to an enhanced multiple reader. To convert a set of storage control names, use the following steps:
1. Issue an XSTART command.
2. Issue an XADDPAIR command for the XRCUTL BB with the SCSESSION id, AA. Repeat the step for all the existing SCSN utility volumes.
3. Issue an XADDPAIR SUSPENDED command to resynchronize the remaining volumes.

If the XRC session is active, and there are multiple SCSNs, use the following steps:
1. Issue an XADDPAIR command for the XRCUTL AA. XRCUTL AA is converted to an enhanced primary session.
2. Issue an XADDPAIR command for the XRCUTL volume, BB, with the SCSESSION id, AA. Repeat the step for all the existing SCSN utility volumes.

Enabling the XRC enhanced multiple readers through the parmlib settings

Refer to the following scenarios to enable the XRC enhanced readers through the parmlib settings.

If there is only one SCSN for one SSID, specify the parameters for the XRC enhanced multiple reader in the parmlib. The SDM determines the conversion to enhanced multiple reader. There must be auxiliary storage control sessions for the SSID.

If there are multiple storage control session names for one SSID with XRCUTL volumes, enable the XRC enhanced readers according to the following steps:

1. Issue an XADDPAIR command for the XRCUTL volume for one of the SCSESSIONs. The session is converted to the enhanced primary mode.
2. Issue an XADDPAIR command for the new XRCUTL volume with the existing SCSESSION id.
3. Specify the PARMLIB setting to indicate how many storage control sessions are used.
4. Remove extraneous XRCUTL devices.

Here is an example for the multiple SCSNs scenario:

Specify 4 read tasks for the following combined SCSNs:

- XADD sessid V(UTC000, XRCUTL) SCSESSION(AA) (convert the first scsession to the enhanced primary)
- XADD sessid V(UTC001, XRCUTL) SCSESSION(AA) (merge the second scsession to the enhanced auxiliary)

1. Issue the following command:
   ```
   XADD(UTC001, XRCUTL) SCSESSION(AA)
   ```
2. Set the parmlib value to 4 and activate the value.

All the volumes in AA and BB on the LSS are in a single session using four reader tasks on volumes UTC000 and UTC001. Two additional volumes are added. You can remove the UTC001 device by issuing an XDELPAIR command.

If there are multiple SCSNs for one SSID without XRCUTL volumes, use the following steps to enable the enhanced multiple reader function:

1. Add an XRCUTL volume to one of the extra SCSNs.
2. Issue the following command:
   ```
   XADD(newvol,XRCUTL) SCSESSION(SCSN id)
   ```
3. Delete the XRCUTL volume from the primary SCSN.

   Note: To merge SCSNs into the primary volume, repeat step 1 to 3.
4. Specify the PARMLIB setting and remove the extraneous XRCUTL volumes.

If there are no volumes available to use as an XRCUTL volume, temporarily stop mirroring at least one volume per LSS. For example, issue an XDELPAIR command, to delete an existing volume pair, to use the pair as the XRCUTL volume. When the XRC enhanced multiple reader starts, issue an XADDPAIR command to add the original volume pair back into the session. If applications are quiesced during the operation, and the original primary and secondary volumes are still synchronized, you can issue a NOCOPY command. If the primary volumes and secondary volumes are different, issue a full COPY command.

Disabling the XRC enhanced multiple reader functions

To disable the XRC enhanced multiple reader function, use the following steps:

1. Issue an XDELPAIR command to deleted the enhanced volumes. If all volumes in the session are using the enhanced multiple reader function, you can issue an XEND command to delete all volumes.
2. Remove the parmlib setting for the NumberReaderTasks parameter.
3. Specify the AllowEnhancedReader parameter value into **No**.

4. Issue an XSTART command.
5. Issue an XADDPAIR command using the non-enhanced configuration. To avoid the delay of full initialization, stop all updates to the primary volumes during this procedure and use the NOCOPY option of XADDPAIR.

   Note: If you stop all updates to the primary volumes, the updates to the primary volumes are lost when XRC is inactive.

Increasing and decreasing the number of read tasks

The following guidelines direct how to increase the number of read tasks:

- If you enable the enhanced multiple reader function through the XRCUTL devices, the number of utility volumes specified in the XADDPAIR command must match the number of read tasks.
- If you enable the enhanced multiple reader function through the parmlib, update the parmlib member and activate the member.

Decreasing the number of read tasks is only supported if all volumes in the session are suspended. The following guidelines direct how to decrease the number of read tasks:

- If you enable the enhanced multiple reader function by using the XRCUTL devices, the number of utility volumes specified in the XDELPAIR command must match the number of read tasks.
- If you enable the enhanced multiple reader function by using the parmlib, update the parmlib member and activate the member.

Adjusting multiple utility volumes and alias addresses

This topic describes how to modify existing enhanced multiple reader configurations to use alias addresses instead of unique volumes, or use volumes instead of alias addresses. To use alias addresses, set the NumberReaderTasks parameter to be a nonzero value. The following scenarios direct how to modify the configurations. The NumbeerReaderTasks value is set to four in all the configurations.

- Configuration one: All independent base volumes, no aliases. The volumes are UTL001, UTL002, UTL003, UTL004.
- Configuration two: Two base volumes, each with one alias (or two shared HyperAV aliases). The volumes are UTL001, UTL002.
- Configuration three: One base volume and three aliases. The volume is UTL001.

To modify configuration two and three to configuration one for using base volumes instead of utility volumes, do the following:

1. Issue an XADDPAIR command to add extra base utility volumes. XRC will use these new utility volumes instead of using alias devices. Message ANTA8171I is issued to reflect that the readers have been re-balanced.
2. Issue XSUSPEND VOLUME(ALL) to suspend all the volumes in the session.
3. Set the NumberReaderTasks value to zero to prevent the use of alias addresses when the base address is busy.
4. Issue XSET PARMLIB to activate the NumberReadrTasks value of zero. This automatically re-adds the base utility or utilities and deletes all of the alias utilities.
5. Issue XADDPAIR SUSPENDED to resume XRC mirroring.

To modify configuration one to configuration two or three for using alias addresses, do the following:

1. Configure alias addresses.
2. Set the NumberReaderTasks value to four.
3. Issue an XDELPAIR command to delete the extraneous utility volumes.

To modify configuration two to configuration three for continuing the use of alias addresses, do the following:

1. Ensure the UTL001 volume has at least three alias addresses.
2. Issue an XDELPAIR command to delete the UTL002 volume.

To change the configuration three to the configuration two for adding a base address, do the following:

1. Ensure UTL002 has at least one alias.
2. Issue an XADDPAIR command against UTL002.
3. Decrease the number of aliases for UTL001.

Chapter 4. Setting up the extended remote copy environment

This section describes how to set up the extended remote copy environment. It explains the system components that interact with XRC, and describes how to do the following:

Note: If you are considering a CXRC environment, see Chapter 8, “Managing coupled extended remote copy sessions,” on page 207 before continuing with this chapter.

In this topic

This topic is intended to help you setup and install extended remote copy. It includes the following sections:

Installing and configuring XRC

About this task

To install and configure the extended remote copy support, perform the following steps after installing the appropriate z/OS release:

Procedure

1. Authorize XRC TSO commands by adding the command names to the AUTHCMD PARM parameter of the IKJTSO*xx* member of SYS1.PARMLIB. See Chapter 5, “Extended remote copy command descriptions,” on page 69 for specific XRC TSO commands. After adding the XRC command names to the IKJTSO*xx* member, issue the TSO command PARMLIB UPDATE(*xx*) to activate the new IKJTSO*xx* member.

   To restrict XRC commands to certain users, follow the instructions in “Controlling access to XRC commands” on page 51.
2. Based on the application or applications that you plan to copy, determine all primary (source) and secondary (target) devices.
3. Allocate the appropriate journal, control, and state data sets on disk devices that have connectivity to the system data mover (SDM) and either the recovery

or the target migration system. You can change the default SYS1 high-level qualifier (HLQ) in any of the examples in Table 13 to a name that corresponds with the HLQ issued on the XSTART command or in the ANTXIN00 PARMLIB member.

Table 13. Data Set Allocation Command Examples.

Command . . .	Notes . . .
SYS1.XCOPY.*session_id*.CONTROL	This is only required with SESSIONTYPE(XRC).
SYS1.XCOPY.*session_id*.STATE	
SYS1.XCOPY.*session_id*.JRNL01	This is only required with SESSIONTYPE(XRC).
SYS1.XCOPY.PARMLIB	This is only used with PARMLIB support.
SYS1.XCOPY.*session_id*.JRNL*xx*	This is only required with SESSIONTYPE(XRC). A minimum of two journals are required (JRNL01 and JRNL02), up to JRNL16.

Note: Ensure that the ANTAS*nnn* address space has update authority to these data sets. See "Address Spaces for XRC" on page 29 for more information on address spaces.

If the XRC system data mover will run on a system that does not share a common time reference with the application systems that will write to the XRC primary volumes, be sure to configure the system in a way that avoids the introduction of incorrect timestamps into the XRC storage control sessions. See "Recovery system clock considerations" for more information.

4. Issue the XSTART TSO command on the system that contains the system data mover. Specify the session ID that is associated with the journal data set names. Select the proper level of recovery for your environment. See "XSTART–Starting a session" on page 110.
5. Issue the XSET commands on the SDM system to tune the environment as you desire. See "XSET–Changing session parameters" on page 102.
6. Issue the XADDPAIR TSO commands on the system data mover to perform the copy of primary to secondary volumes. See "XADDPAIR–Adding volume pairs or utility volumes" on page 71.
7. Issue the XQUERY TSO command on the SDM system to verify the XRC installation options currently in effect. See "XQUERY–Querying a session" on page 90.
8. If you are using XRC for disk devices migration, follow the disk migration scenario that is described in Chapter 10, "Migrating data with extended remote copy," on page 251.
9. If you are using XRC for disaster recovery, XRC continues to copy updates from the primary volumes to the secondary volumes for as long as the system is running, or until you issue a command to end the volume copies.

Recovery system clock considerations

Timestamping of write operations is an essential mechanism that XRC uses to maintain sequence consistency of write operations to the secondary volumes. When the XRC system data mover is run on a remote recovery system that does not share a common time reference with the application systems, it is important to configure the system in a way that avoids the introduction of incorrect timestamps into the XRC storage control sessions. To do this, you should use SuppressTimestamp(YES) in parmlib member ANTXIN00. For more information, refer to "ANTXIN00 parmlib parameters" on page 126..

Virtual storage messages

If you receive pageable-storage shortage messages, the SDM system is constrained by lack of real or virtual storage. In this case, either reduce the number of primary storage controls being managed by the SDM or increase the amount of real storage or paging space for the ANTAS*nnn* address space. You can also reduce the maximum number of buffers the data mover can use. This is done by setting the TotalBuffers parameter to a smaller value. Each buffer requires 60K of virtual and real storage.

For additional information about address spaces, refer to "Address Spaces for XRC" on page 29.

Controlling access to XRC resources

This section describes how to protect XRC commands and control volume repair activities.

Controlling ICKDSF activity to XRC volumes

You can use ICKDSF to change the VOLSER of volumes that are active under XRC. Messages ANTI8027I, ANTI8029I, and ANTI8037I are issued to explain the changes that have taken place.

Using ICKDSF to manage LOGPLUS volumes

The ICKDSF volume utility REFORMAT command with the XRCLOGGER parameter, or the INIT command, can be used to manage LOGPLUS designated volumes. The ICKDSF volume utility may be needed if a volume state is left as L+ even after Logger is no longer actively using the volume. Refer to *Device Support Facilities (ICKDSF) User's Guide and Reference* for more information.

Controlling access to XRC commands

The following are two ways to protect access to XRC commands:

- Put the commands in a library that is protected by RACF.
- Define resource profiles in the RACF facility class and restrict access to those profiles.

Placing TSO commands in an RACF-protected library

Place the TSO commands in a RACF-protected library to restrict XRC TSO commands to authorized storage administrators.

To RACF-protect XRC commands, perform the following steps:

1. Issue the following RDEFINE command for each XRC command, and for each command abbreviation that you want defined to RACF:

   ```
   RDEFINE PROGRAM cmdname ADDMEM('SYS1.CMDLIB')/volser/NOPADCHK) UACC(NONE)
   ```

   The following terms apply to the above example:

   cmdname
   : Defines the XRC TSO command name or an abbreviation of a

command. Issue a separate RDEFINE command for each of the XRC commands, and any command abbreviations you plan to use. Examples of XRC command abbreviations are XADD, XDEL, and XRCV. RACF can only perform checking on commands and abbreviations that are defined to it.

volser
: Defines the name of the volume that contain the SYS1.CMDLIB data set.

2. Issue the PERMIT command for all commands and authorized XRC TSO command users as follows:

   ```
   PERMIT cmdname CLASS(PROGRAM) ID(name) ACCESS(READ)
   ```

   The following terms apply to the above example:

   cmdname
   : Defines the XRC TSO command name, or an abbreviation of a command.

   name
   : Defines the user ID receiving RACF access authority for that command name.

3. Issue the SETROPTS command from a user ID that has the appropriate authority:

   ```
   SETROPTS CLASSACT(PROGRAM) WHEN(PROGRAM) REFRESH
   ```

Defining resource profiles in the RACF FACILITY class

You can limit the use of XRC commands by defining resource profiles in the RACF FACILITY class and restricting access to those profiles. To use a protected command, you need read-access authority to the applicable profile.

Table 14 lists the XRC commands and the FACILITY class profiles that can restrict them.

Table 14. FlashCopy and XRC FACILITY class profile names

Command	Profile name
FCESTABL	STGADMIN.ANT.XRC.COMMANDS
FCWITHDR	
XADDPAIR	
XADVANCE	
XCOUPLE	
XDELPAIR	
XEND	
XQUERY	
XRECOVER	
XSET	
XSTART	
XSUSPEND	
FCQUERY	STGADMIN.ANT.XRC.XQUERY
XQUERY	
XSTATUS	

Table 14. FlashCopy and XRC FACILITY class profile names (continued)

Command	Profile name
Note: Authorize FCQUERY and XQUERY command use with the STGADMIN.ANT.XRC.COMMANDS profile or the STGADMIN.ANT.XRC.XQUERY profile. XRC first checks STGADMIN.ANT.XRC.COMMANDS for authorization. If authorization is not permitted with the STGADMIN.ANT.XRC.COMMANDS profile, XRC checks the STGADMIN.ANT.XRC.XQUERY profile for authorization to issue the FCQUERY and XQUERY commands.	

Examples: The following examples activate the RACF FACILITY class, define the profile for the XRC commands, and give user STGADMIN authority to use this profile.

- Activate the RACF facility class:

```
SETROPTS CLASSACT(FACILITY)
```

- Define the profile for XRC commands, and authorize user STGADMIN to use this profile:

```
RDEFINE FACILITY STGADMIN.ANT.XRC.COMMANDS UACC(NONE)

  PERMIT STGADMIN.ANT.XRC.COMMANDS CLASS(FACILITY) -
         ID(STGADMIN) ACCESS(READ)
```

For additional information about how to activate the RACF FACILITY class and how to define and authorize users to the XRC command profiles, refer to *z/OS Security Server RACF Security Administrator's Guide*.

Including XRC TSO commands in automated procedures

You can include XRC TSO commands in automated procedures, such as REXX execs. Within the automated procedure, use the MSGROUTEID parameter to specify the TSO user ID to which you want XRC to send messages.

XRC routes operational messages to the user ID that issued the command. In order to have these messages appear in the SYSLOG, ensure that the SYS1.PARMLIB CONSOL*xx* member permits “Write to Programmer” on the ROUTCODE parameter.

Using the ANTRQST API to run XRC

You can program XRC calls to the z/OS SDM application system data mover programming interface. Appendix C, “ANTRQST and ANTRQSTL macros – call to the system data mover API,” on page 573 contains information on the ANTRQST environment, programming requirements, common invocation parameters, input and output register information, followed by the macro syntax and individual parameter descriptions. An example of the ANTRQST macro in assembler follows the descriptions of the associated return and reason codes.

Specifying XRC journal, control, and state data sets

This section describes the XRC journal, control, and state data sets. These data sets can reside on any device, and can be attached to any storage control that supports the sequential access method (SAM).

If these data sets are Resource Access Control Facility (RACF) protected, ensure that the ANTAS*nnn* address space, which updates them, has ALTER access authority on both the SDM and recovery systems. RACF protects these vital data sets and ensures that no changes are made to them whenever XRC is not active.

Before activating XRC, allocate and catalog the necessary support data sets on volumes that both the SDM and the recovery system host can access. Table 15 shows which data sets should be allocated with each command.

Table 15. Data set allocation guide

For . . .	Allocate . . .
a SESSIONTYPE of MIGRATE	the state data set.
a SESSIONTYPE of XRC	the journal, control, and state data sets.
CXRC	the master data set.
Clustering	the cluster data set and the master data set.
Cluster monitoring	the data set for clustering and the cluster state data set.

Note: Because the control and state data sets must be SMS-managed, use the IDCAMS recatalog function when you activate the recovery system if these data sets do not already reside there.

Guidelines: Follow these guidelines when defining journal, control, and state data sets.

- The state and control data sets can reside on the same volume, but neither should be on any volume that also contains journal data sets. Because the performance of the journal data set and control data set is so important, be sure to follow the guidelines that are described in "Maximizing access to journal data sets" on page 67.
- To avoid a potential deadlock situation, place the XRC state, control, and journal data sets in a user catalog that contains only entries for these data sets. If you plan to copy the volume that contains these data sets, ensure that the volume does not also contain data sets that application programs are updating. Either of these conditions can also cause a deadlock condition when XRC sessions are suspended or restarted.

XRC cannot dynamically accept changes to the journal, state, or control data sets. If you wish to change these data sets, you must follow the instructions listed in Table 16 on page 55.

Table 16. Instructions for changing journal, control, and state data sets

To change . . .	Choose . . .	Follow these directions . . .
Journal or control data sets		1. Issue the XSUSPEND command with the TIMEOUT option to suspend the session if it is still active. 2. If the session is coupled, issue the XCOUPLE command with the PURGE option to uncouple it. 3. Restart the session using the XSTART command with SESSIONTYPE(MIGRATE). 4. Again, issue the XSUSPEND command with the TIMEOUT option to suspend the session. 5. Delete the old data set(s) and allocate the new ones. 6. Restart the XRC session using the XSTART command with SESSIONTYPE(XRC). 7. Perform resynchronization using the normal procedures for your installation.
State data sets	to end the session and reallocate the data set. **or**	1. Issue the XEND command. 2. Reallocate the state data set. 3. Reissue the XSTART and subsequent commands.
	to suspend the session and allocate a new data set.	1. Issue the XSUSPEND command. 2. Allocate a new state data set. 3. Copy the suspended state data set members into the new state data set. 4. Delete the original state data set. 5. Rename the new state data set to the original state data set name. 6. Restart the session by issuing XSTART and subsequent commands.

For additional information about address spaces, refer to "Address Spaces for XRC" on page 29.

Specifying the journal data sets

Journal data sets contain checkpoint records of the changes that are made, and are read only when you issue an XRECOVER command.

XRC uses a journal data set name in the form:

SYS1.XCOPY.*session_id***.JRNL***nn* or *hlq***.XCOPY.***session_id***.JRNL***nn*

Example: The following is an example of a journal data set name:

```
SYS1.XCOPY.DALLAS.JRNL01
```

Required specifications

The following requirements are necessary when you specify journal data sets:

- Allocate journal data sets on disk volumes. Each journal data set must be a fixed-block SAM data set with the following attributes:

```
DCB=(RECFM=FB,LRECL=7680,BLKSIZE=7680,DSORG=PS),SPACE=(TRK,(1200,0))
```

  Note: If the LRECL and BLKSIZE are not set to 7680, the system data mover will attempt to force the allocation to this size.
- Specify at least two journal data sets. If you allocate less than the maximum number of journal data sets, XRC issues the following message:

```
DATA SET SYS1.XCOPY.session_id.JRNLnn NOT IN CATALOG OR CATALOG
     CANNOT BE ACCESSED
```

  A high-level qualifier name can replace SYS1 in the example. XRC issues the above message for the first journal data set that is not found, even if there are two or more journal data sets allocated.
- Do not use compression algorithms on journal data sets.
- Observe the following when naming journal data sets:
  - Specify up to eight alphanumeric characters for the *session_id* portion of the journal data set name. Use the same name that you will specify with the *session_id* parameter of the XSTART command.
  - Name the journal data sets contiguously with a number, *nn*, beginning at 01 up to a maximum of 16.

    Example: If you define journal data sets 01, 02, and 04, XRC ignores data set 04 because you did not define data set 03.

Recommended specifications

Use the following steps as guidelines to help to improve the efficiency of your XRC operations:

- Allocate journal data sets as single-extent data sets. If secondary extents are used, the data set must be filled prior to starting the XRC session if you want XRC to use the full amount of space that you have allocated for journal use.
- Define an even number of journal data sets, preferably four to eight for smaller XRC configurations (up to about 750 volumes). For large XRC configurations, define from eight to 16 journal data sets.
- Define the journal data sets as striped SAM data sets.

  When you define the journal data sets as striped SAM data sets on different volumes, you enable the system data mover to spread its I/O operations across multiple volumes and storage controls. Spreading the work load reduces contention for the same physical device. If you use nonstriped journal data sets, each data set must reside on a single volume.
- Make journal data sets large enough to allow the system data mover to offload data to secondary volumes during peak data load periods. See Note.
- Do not put the control data set on any of the journal data set volumes.
- Minimize non-XRC activity to the journal data set volumes.
- Do not allocate the journal data sets in the Extended Address Space (EAS) of Extended Address Volumes (EAV) unless you are certain that all SDM LPARs that may potentially require access to the journals (for example, from all potential active and recovery systems) are at z/OS V1R11 or higher.

Note: To maximize performance and avoid stall conditions, allocate each journal data set so that it can hold data for every buffer that the session might use. This includes the number of buffers specified in the TotalBuffers value, plus an additional 20% for "Emergency buffers". Also, when determining the number of tracks required for a journal dataset, allow four journal tracks for every three buffers to be stored. This translates to a requirement of 1.6 journal tracks for each buffer. So, for example, an installation that uses TotalBuffers(25000) should allocate each journal data set with a total primary allocation of 25,000 * 1.6 = 40,000 tracks.

Optimal performance specifications

The following steps are recommended to achieve optimal journal performance:

- Allocate the journal data sets on high-performance volumes, and place them behind storage controls that use DASD fast write, cache functions, and have large NVSs.
- Attach the journal data sets with high bandwidth ESCON or FICON channels, preferably capable of very high transfer rates.
- Allocate the same amount of space for all journal data sets.
- Put the journal data set volumes behind storage controls that do not also process primary or secondary volumes.
- Spread the striped journal data sets across as many disk volumes as possible, preferably with each journal data set on a separate volume. Processing performance is improved because XRC assigns a separate internal subtask for each pair of allocated journal data sets.
- For a fully-configured, active mirror do not use less than four journal data sets.

Table 17 and Table 18 on page 58 show examples of journal data set patterns for two and eight stripes across four, eight and sixteen journal data sets.

Table 17. Sample Journal Data Set Patterns for Two Stripes

Journal Data Sets	Striped data sets
Two	VOL1 VOL2 JRNL01 JRNL01 JRNL02 JRNL02
Four	VOL1 VOL2 VOL3 VOL4 JRNL01 JRNL01 JRNL03 JRNL03 JRNL02 JRNL02 JRNL04 JRNL04
Eight	VOL1 VOL2 VOL3 VOL4 JRNL01 JRNL01 JRNL03 JRNL03 JRNL02 JRNL02 JRNL04 JRNL04 VOL5 VOL6 VOL7 VOL8 JRNL05 JRNL05 JRNL07 JRNL07 JRNL06 JRNL06 JRNL08 JRNL08

Table 17. Sample Journal Data Set Patterns for Two Stripes (continued)

Journal Data Sets	Striped data sets
Sixteen	VOL1 VOL2 VOL3 VOL4 JRNL01 JRNL01 JRNL03 JRNL03 JRNL02 JRNL02 JRNL04 JRNL04 VOL5 VOL6 VOL7 VOL8 JRNL05 JRNL05 JRNL07 JRNL07 JRNL06 JRNL06 JRNL08 JRNL08 VOL9 VOL10 VOL11 VOL12 JRNL09 JRNL09 JRNL11 JRNL11 JRNL10 JRNL10 JRNL12 JRNL12 VOL13 VOL14 VOL15 VOL16 JRNL13 JRNL13 JRNL15 JRNL15 JRNL14 JRNL14 JRNL16 JRNL16

Table 18. Sample Journal Data Set Patterns for Eight Stripes

Journal Data Sets	Striped data sets
Four	Vol 1 Vol 2 Vol 3 Vol 4 JRNL01 JRNL01 JRNL01 JRNL01 JRNL02 JRNL02 JRNL02 JRNL02 Vol 5 Vol 6 Vol 7 Vol 8 JRNL01 JRNL01 JRNL01 JRNL01 JRNL02 JRNL02 JRNL02 JRNL02 Vol 9 Vol 10 Vol 11 Vol 12 JRNL03 JRNL03 JRNL03 JRNL03 JRNL04 JRNL04 JRNL04 JRNL04 Vol 13 Vol 14 Vol 15 Vol 16 JRNL03 JRNL03 JRNL03 JRNL03 JRNL04 JRNL04 JRNL04 JRNL04

For additional information about optimizing XRC performance, refer to "Maximizing access to journal data sets" on page 67.

Specifying the control data set

The recovery operation uses the control data set to determine what data XRC must still write to the secondary volumes. The control data set contains control records that indicate:

- The last set of data that is written to the secondary volumes
- The amount of unwritten data that exists in the journal
- The location of this unwritten data

XRC uses a control data set name in one of the following forms:

```
SYS1.XCOPY.session_id.CONTROL  or  hlq.XCOPY.session_id.CONTROL
```

The XRC session ID can be up to eight characters long and must be the same name that you will specify with the *session_id* parameter of the XSTART command.

Example: The following is an example of a control data set name:

```
SYS1.XCOPY.DALLAS.CONTROL
```

Guidelines: Use the following guidelines to allocate the control data set:

- Place the control data set on a different volume from the journal data sets, ideally on its own volume.
- Allocate the data set with physical sequential organization, and without extended format.
- Prior to its first use, initialize the data set with IEBDG. This is sample JCL for allocation and initialization:

```
//STEP1    EXEC PGM=IEBDG
//SYSPRINT DD SYSOUT=*
//DD1      DD DISP=(NEW,CATLG),
//            DSN=SYS1.XCOPY.TPC.CONTROL,
//            STORCLAS=(JRNLSTD),
//            DCB=(RECFM=FB,LRECL=15360,BLKSIZE=15360,
//            DSORG=PS),SPACE=(TRK,(1,0))
//SYSIN    DD  *
  DSD    OUTPUT=(DD1)
  CREATE QUANTITY=2
  END
/*
```

- Do not allocate the control data set in the Extended Address Space (EAS) of an Extended Address Volume (EAV).

Notes: Any valid combination of RECFM, LRECL and BLKSIZE that results in two single-record blocks on the track may be used. However, the system data mover uses EXCP to write the data set, and may write blocks of different size than you specify. Therefore, you will not be able to browse or otherwise process the dataset with standard access methods once it has been used in an XRC session. The DCB attributes shown in the sample JCL have been chosen to create this situation, and discourage editing or other processing of the data set.

Once a control data set has been used in an XRC session, do not attempt to initialize it again. Should the data set become corrupted, you will not be able to XRECOVER or XADVANCE the session, nor will you be able to restart a coupled session. Restart is possible only by first uncoupling the session through the XCOUPLE PURGE command.

XRC also supports allocation of a PDSE control data set with LRECL=4096 and BLKSIZE=4096. However, IBM recommends sequential allocation as it provides much higher performance, and is also required if Coupled XRC is to be used.

Specifying the state data set

The state data set defines and contains the status of the XRC session and of associated volumes pairs that XRC is currently managing. The state data set is updated whenever an XADDPAIR, XDELPAIR, XSET, XSUSPEND, XRECOVER, or XEND command is issued, or whenever a volume state changes.

XRC uses a state data set name in one of the following forms:

```
SYS1.XCOPY.session_id.STATE  or  hlq.XCOPY.session_id.STATE
```

Rule: The XRC session ID can be up to eight characters long and must be the same name that you will specify with the *session_id* parameter of the XSTART command.

Example: The following is an example of a state data set name:

```
SYS1.XCOPY.DALLAS.STATE
```

Guidelines: The following are guidelines for the state data set:

- Allocate the state data set on disk as an SMS-managed partitioned data set extended (PDSE) data set with the following attributes:

```
DCB=(RECFM=FB,LRECL=4096,BLKSIZE=4096,DSORG=PO),DSNTYPE=LIBRARY
```

- Do not allocate the state data set as a generation data set (GDS) PDSE. If you do allocate the state data set as a GDS PDSE, it might cause unpredictable results and interfere with the use of XRC Performance Monitor (XPM) or other utility programs that access the STATE data set.
- Allocate ten tracks per storage control session. Try to plan for expected future growth when you initially allocate the state data set, as it may be inconvenient to reallocate it later.
- For non-ESS storage controls, allocate one track for each volume pair in the storage control session and try to plan for the expected future growth. For volumes that have a capacity larger than a 3390 Model 3 (3339 cylinders), allocate one additional track for every 3000 additional cylinders. Ensure that the allocation size is sufficient because XRC posts an error if the state data set runs out of space. You cannot add additional storage control sessions or volume pairs if there is insufficient space in the state data set. When there is insufficient space in the state data set, you cannot couple a session to a master session.

  Use the following steps to allocate a new, larger data set:

  1. Issue XSUSPEND *session_id* TIMEOUT (*hh.mm.ss*).
  2. Copy all members from the existing state data set to the new, larger data set.
  3. Rename the existing state data set to an unused name.
  4. Rename the new, larger data set to the pre-existing state data set name.
  5. Issue XSTART and resume the XRC operations.
- Do not allocate the state data set in the Extended Address Space (EAS) of an Extended Address Volume (EAV) unless you are certain that all SDM LPARs that may potentially require access to the state data set (for example, from all potential active and recovery systems) are at z/OS V1R12 or higher.

Note: Some environments can benefit from specifying a STORCLAS parameter to ensure that SMS allocates the state data set to the proper SMS-managed volume.

For additional information about the syntax for specific commands, refer to Chapter 5, “Extended remote copy command descriptions,” on page 69.

Specifying the cluster data set

The cluster data set (CDS) records status on disk, for use in restart situations. Data set attributes are the same as a master control data set. You must create the cluster data set before you enable clustering for a logical partition.

XRC uses a cluster data set name in the following form:

```
mhlq.XCOPY.ClusterName.CLUSTER
```

The *ClusterName* can be up to eight characters long and must be the same name specified on the XRC parmlib parameter of *ClusterName* under the STARTUP category.

Guidelines: Use the following guidelines to allocate the cluster data set:

- Place the cluster data set on a different volume from the journal data sets, ideally on its own volume.
- Allocate the data set with physical sequential organization, and without extended format.
- Before the first use of a cluster data set, initialize the data set with IEBDG.
- Do not allocate the cluster data set in the Extended Address Space (EAS) of an Extended Address Volume (EAV).

Specifying the cluster state data set

The cluster state data set records accumulated XRC performance monitor information for all XRC sessions coupled througha cluster session. Member MONITOR1 contains this accumulated information. The XRC Performance Monitor uses this information. The attributes for this data set are the same as a STATE data set for an XRC session. If the data set is not created before an attempt is made to enable clustering for an LPAR, then clustering is allowed to proceed, with the monitoring function disabled. This data set is optional.

XRC uses a Cluster state data set name in the following form:

```
mhlq.XCOPY.ClusterName.STATE
```

Rule: The *ClusterName* can be up to eight characters long and must be the same name specified on the XRC parmlib parameter of ClusterName under the STARTUP category.

Guidelines: Use the following guidelines to allocate the cluster state data set:

- Allocate a 2 - track SMS managed partitioned data set extended (PDSE) data set with the following attributes:

```
DCB=(RECFM=FB, LRECL=4096, BLKSIZE=4096,DSORG=PO), DSNTYPE=LIBRARY
```

- Do not allocate the cluster state data set in the Extended Address Space (EAS) of an Extended Address Volume (EAV).

Creating a PARMLIB data set

The main purpose of creating a PARMLIB data set is to provide a single method to specify parameters for each system in a multisystem environment, without modifying the ANTXIN00 member of SYS1.PARMLIB. XRC uses a PARMLIB data set name in one of the following forms:

hlq.**XCOPY.PARMLIB**

or

SYS1.XCOPY.PARMLIB

Guideline: Allocate *hlq*.XCOPY.PARMLIB as RECFM fixed, with LRECL=80 and DSORG=PO.

For additional information about the parameters you can specify in the PARMLIB data set, refer to Chapter 6, "Administering your extended remote copy environment," on page 123.

Copying the catalog and control data sets

You can ensure faster disaster recovery if you use remote copy functions to copy volumes that contain the master catalog, key user catalogs, and system control data sets to the recovery system.

Remote copy automatically copies all data sets on the managed volumes. Managed volumes can include catalog data sets and control data sets. XRC can copy *any* data set type, which includes JES spool, JES checkpoint, system volumes, and paging volumes.

Specifying XRC utility volumes

XRC utility volumes are specified according to the configuration of the storage subsystem. See Table 19 when specifying XRC utility volumes. For more information, see "Using XRC utility devices" on page 165.

Table 19. Decision process for utility volume specification

If . . .	Then . . .	Notes . . .
Operating with non-IBM ESS storage subsystems:	Specify at least one (ideally more than one) low-activity volume for each application (primary) storage control session as a utility volume (or volumes) for the system data mover.	• If multiple volumes exist in the storage control session, they need not be dedicated, as the SDM switches to the least-busy volume at different times. SDM performance improves when you specify several candidate volumes that have little or no activity. This, in turn, improves the application I/O performance to volumes in XRC sessions. • See "Using XRC utility devices" on page 165 for more information.

Table 19. Decision process for utility volume specification (continued)

If . . .	Then . . .	Notes . . .
Operating with the ESS storage subsystem:	You have the option to define a single-cylinder volume for specific use as a utility volume.	See "Using XRC utility devices" on page 165 for more information.

Identifying volume pairs

The storage administrator must identify the application volumes to copy, and create extended remote copy volume pairs. XRC provides an image copy of a volume on a record-for-record basis. There is a one-to-one correspondence between the record on the primary volume track and the record on the secondary volume track. Because the data written on the primary disk is written to the same tracks on the secondary disk, the secondary disk must have the same track size and number of tracks per cylinder, and the same or larger volume capacity, as the primary. (If these criteria are not met, the XADDPAIR command will fail.) Ensure that secondary volumes are part of a storage group that does not allow allocations, and dedicate these volumes to XRC use only.

Remote copy supports volumes, not data sets. As a result, all data sets on the volumes that are copied are part of remote copy activity, and are therefore copied to the target volume. This support is application (IBM or non-IBM) independent, and supports all data set types.

Because applications deal with data sets and not volumes, multivolume data sets require special attention. Multivolume data set types include data sets that reside on multiple volumes, striped data sets, and VSAM spheres. Unless you copy all volumes of a multivolume data set, XRC will copy only part of the data set. Other data sets on the copied volume may be usable, but the multivolume data sets will not be. If multivolume data sets are critical for recovery, you must copy all volumes on which these data sets reside. In addition, if these data sets are critical to the system, start the XRC session by issuing the XSTART or XADDPAIR command with the ERRORLEVEL(SESSION) or (*group_name*) parameter specified.

In the event of an error, specifying ERRORLEVEL(SESSION) ensures data consistency across all volume pairs in the entire XRC session. Specifying ERRORLEVEL(*group_name*) ensures that data is consistent across all volume pairs that belong to the specified group. Errors within the specified group do not affect the update activity of volume pairs that belong to other groups.

If the data sets require consistency as a group, issue the XADDPAIR command for these related data sets. Specify the ERRORLEVEL parameter with a group name that is meaningful for the related volumes. Specified as such, an error for one volume pair would suspend the entire set of volumes that are associated with the named group. After correcting the error, you can add the volume group back to the XRC session.

For additional information about the XADDPAIR command, refer to "XADDPAIR–Adding volume pairs or utility volumes" on page 71.

For additional information about the ERRORLEVEL parameter, refer to Chapter 11, "Recovering from error conditions using extended remote copy," on page 255.

Optimizing XRC performance

The information in this section adds to the basic guidelines that are presented in "Specifying XRC journal, control, and state data sets" on page 54. See your IBM marketing representative for the latest remote copy performance and configuration information and recommendations for your specific environment.

Ideally, for the best XRC system performance and recoverability, the recovery system would be equivalent to the primary system. In reality, you may only need to back up a subset of your primary system as a result of a disaster. The following sections outline additional ways to optimize overall XRC system performance.

Configuring for high performance

Extended remote copy works together with cached storage controls to efficiently manage existing system resources for disaster recovery. The system data mover issues I/Os to drain record updates from the cache that are designated to be copied by XRC. Each logical XRC session may have one or more storage control sessions for each primary storage control. A logical XRC session manages all volumes that are associated with those storage control readers.

High performance can be achieved by ensuring that the storage subsystems have adequate resources, including cache, NVS, and path connections to handle the peak workload that is associated with remote copy sessions (and coexisting concurrent copy operations). The SDM software requires adequate processor MIPs, multiple processors, and real storage to accomplish its function in an efficient and timely fashion. The SDM, without adequate resources, may not be able to drain the storage control cache rapidly enough under peak workload conditions. Based on the workload, if this continues for an extended period of time, the cache could become overcommitted, and ultimately this could affect the performance of your primary systems.

Given adequate resources, the system data mover functions smoothly even as the workload stress on it grows. Constrained resources will become apparent as they create bottlenecks that impede the performance of the system data mover.

Using parallel access volumes for secondary devices

Enabling parallel write I/O for XRC secondary devices can improve overall data mover throughput and reduce average session delay. To enable parallel writes, the secondary logical subsystems must be configured with parallel access volumes (PAVs). Static aliases can be used, but for optimum flexibility IBM recommends using the HyperPAV feature of IBM system storage. The use of the Dynamic Alias Management function of z/OS Workload Manager is not recommended for XRC secondary devices.

Recommendations for controlling the storage control cache

Use the recommendations listed in this section as guidelines to help reduce the likelihood of overcommitting the storage control cache:

Define one or more separate, low-activity volumes: Defining one or more separate, low-activity volumes specifically for system data mover I/O activity is helpful for storage subsystems that do not have the functional capabilities of the ESS storage subsystem. By having a dedicated reader volume, the system data mover can avoid conflicts with application programs that share a utility volume. Sharing a utility volume can negatively impact the performance of both the SDM and the application program.

The XRC system data mover software issues its I/O to the XRC utility volume that is associated with each primary storage control. This is part of the data transfer process. You may assign one or more XRC utility volumes to each primary storage control by specifying the SCSESSION keyword on the XADDPAIR command. Each specified unique SCSESSION will start a separate reader. All volumes in an LSS (or SSID) with the same SCSESSION share the same reader.

For additional information about using XRC utility devices, refer to "Using XRC utility devices" on page 165.

Assign multiple storage control sessions: Assign multiple storage control sessions with the SCSESSION keyword on the XADDPAIR command. Each storage control session has an independent SDM task that is committed to it. The system data mover is thereby capable of reading concurrently from each of the storage control sessions. Although it is possible to combine independent applications into a single session, this increases the SDM's workload. Providing separate storage control sessions for each independent application achieves the best performance.

Table 20 provides a summary of XRC session configurations.

Table 20. Quick reference for XRC session limits

You can configure up to . . .	For each . . .
4 XRC sessions, and up to 64 combined XRC and CC sessions	Primary site 3990 or 9390 Storage Control
64 combined XRC and CC sessions	Primary site ESS logical storage subsystem
256 volumes	ESS logical storage subsystem
128 volumes	3990 or 9390 Storage Control
80 storage control sessions	XRC session
1 XRC session, and up to 16 combined XRC and CC sessions	volume
20 XRC sessions (coupled or uncoupled)	Logical partition when in LPAR mode
14 combined XRC coupled and cluster sessions	master CXRC session
13 XRC sessions (coupled)	cluster session in the logical partition when in LPAR mode

XRC maintains update sequence integrity among the volumes that you assign as part of an MVS XRC storage control session. The SDM, however, cannot maintain update sequence integrity between different XRC sessions unless the different sessions are coupled to the same master session. Therefore, if you are not running CXRC, keep all volumes associated with an application that requires update sequence integrity within the same MVS XRC session.

Use caution when combining both XRC and concurrent copy operations on the same storage control: Use caution when combining both XRC and concurrent copy operations on the same storage control for storage controls that do not support hardware bitmapping.

Concurrent copy and XRC both use storage control cache. If you run both simultaneously, you could deplete the cache resources on storage controls that do not support hardware bitmapping. In this case, the storage subsystem may cancel concurrent copy activity. You may instead want to run concurrent copy during low XRC update activity periods, or else run concurrent copy against the secondary volumes of the XRC pair.

Increase the storage control cache size: You can help manage the peak write activity load for XRC-managed volumes by increasing the size of the storage control cache. Due to the large volume of data being moved, storage controls that support XRC may require significantly more cache memory than do those without this ability. The peak load for the storage control dictates the amount of cache that is required. Storage controls require more cache to prevent canceled XRC sessions as the peak load becomes heavier.

Depending on your application write I/O rate, the primary storage subsystem may require additional cache. Sophisticated applications with high write content can require as much as 6 GB or more of subsystem cache. The actual requirements depend on the application, and on the capability of the storage subsystem.

Increase the SDM transfer performance: Increase the transfer performance by connecting it to the primary storage control with the highest bandwidth channels. If you use channel extenders, ensure that the compression feature is installed, since this will typically double the bandwidth.

Optimize the recovery system setup: Allocate the XRC control, state, and journal data sets, and the secondary volumes behind storage controls that have large cache and NVS storages. This configuration, in addition to speeding the XRC copy process, also provides a powerful platform for the recovery operation. For the best setup, configure the secondary site storage controls to have at least the same cache and NVS capacity as the primary site storage controls. In addition, if you have the journal data sets on an ESS, you may want to place them on volumes that utilize parallel access volumes (PAVs), because the journal data sets will take advantage of this capability.

Note: If the system data mover ends with an ANTX5118E or ANTX5119E message, the I/O rate of the primary subsystem may have exceeded the capacity of the subsystem at the receiving end.

Provide a stand-alone SDM system or LPAR: If possible, place the system data mover on its own system with a large amount of real storage. This ideal setup allows the SDM to allocate a large number of track-size buffers and to optimize its I/O performance with minimal impact to primary application I/O. A separate system also minimizes the impact to the system data mover, eliminating a potential bottleneck due to capacity problems during peak I/O update times.

Assign SDM fixed pages: To take advantage of the available real storage, issue the XSET PAGEFIX command to assign the system data mover 35 MB per defined reader. The ReleaseFixedPages parameter found in parmlib support offers a method to keep the pages fixed.

For additional information about how to use the PARMLIB parameters, refer to "ANTXIN00 parmlib parameters" on page 126.

Provide adequate host resources for the system data mover: XRC copy operations rely on host processing resources. Therefore, analyze the host resources that are necessary for your applications. Ensure that the SDM system has the necessary processing capacity required to maintain optimal performance. This includes defining processors with the appropriate MIP capability (about 2-4 MIPs per 100 write updates) and multiprocessor capability (2-4 processors per SDM). The SDM supports a uniprocessor environment. Multiprocessors allow more parallelism, but if the MIPs are sufficient the data mover runs efficiently on a single processor.

Balancing storage control configurations

Configure your system so that primary system activity does not exceed the capacity of the recovery site storage control.

XRC supports configurations where many primary site storage controls can funnel their updated data to a single storage control on the recovery system. However, you must consider the number of copied primary volumes and the rate of write activity to these volumes in your overall work load evaluation for the configuration. An inadequate configuration can cause XRC to cancel the copy operation to one or more volumes, or even end the storage control session.

Distributing work loads

Avoid directing all update activity to a small set of common volumes on a single recovery site storage control. Typically, only a small number of devices receive most of the activity on a particular storage control. Usually two to four devices account for 50% or more of the total update activity and 8-10 volumes account for 80% of the update activity during any stress timeframe. If, for example, two primary site storage controls channel their most active volumes into a single recovery site storage control, the recovery subsystem could become rapidly overcommitted with copy activity.

Balancing system data mover configurations

XRC supports large configurations where many system data movers are required to handle the whole workload. The system data movers may exist in a coupled XRC master session. Any delays in one session may cause delays in other sessions. To avoid a persistent workload imbalance between data movers, use the following guidelines:

- Balance the workload (highly active volumes, readers, and LSS) across data movers.
- Run the maximum number of data movers that the available resources (processor and memory) on an individual LPAR permits.
- Enable clustering on LPARs and balance data movers across the clusters.
- Balance central electronics complex (CEC) resources across the data mover LPARs.

Maximizing access to journal data sets

About this task

Where and how you allocate journal data sets affects XRC performance, except in migration mode. XRC journals all copied data, and the capacity of the journal data sets must be enough to support the primary system I/O rate. You do not need journal and control data sets when the XRC session is running in SESSIONTYPE(MIGRATE) mode.

XRC can occasionally fill all of the journal data sets, in which case it must wait for secondary update processing to release journal space before continuing. If the full-journal condition is not relieved within a reasonable amount of time, XRC suspends the session, and then restarts it. If this happens, you may need to enlarge the journal data sets. Follow the procedure that is documented in “Specifying XRC journal, control, and state data sets” on page 54 to enlarge the journal data sets.

Perform the following steps to enlarge the journal data sets.

Procedure

1. Issue the XSUSPEND command with the TIMEOUT option to suspend the session if it is still active.
2. If the session is coupled, issue the XCOUPLE command with the PURGE option to uncouple it.
3. Restart the XRC session with the XSTART command with SESSIONTYPE(MIGRATE). (See "XSTART–Starting a session" on page 110).
4. Again, issue the XSUSPEND command with the TIMEOUT option to suspend the session.
5. Enlarge the size of the journal data sets or allocate additional ones.
6. Restart the XRC session using the XSTART command with SESSIONTYPE(XRC).
7. Perform resynchronization using the normal procedures for your installation.

Chapter 5. Extended remote copy command descriptions

This chapter describes extended remote copy (XRC) Time Sharing Option (TSO) commands used for starting and managing your XRC sessions. XRC can be managed through TSO commands and the ANTRQST API described in "ANTRQST application programming interface overview" on page 121.

XRC processes commands in the order that you issue them, except for the XQUERY and XSET commands, which are processed immediately. All XRC TSO commands can be put into a SYS1.PROCLIB member, included within standard job control language (JCL), or put into CLISTs.

In this topic

This topic provides the commands used for administering your XRC environment. The following tasks that are associated with using these commands are described in Chapter 7, "Managing extended remote copy operations," on page 157.

Issuing extended remote copy TSO commands

Extended remote copy TSO commands have the same restrictions as other TSO commands. The TSO abbreviation convention requires that you specify as much of the command name or parameter as is necessary to distinguish it from other command names or parameters. The command syntax diagrams show all unique abbreviations.

You can issue XRC commands to control an XRC session as soon as TSO becomes operational. Table 21 on page 70 lists the XRC TSO commands and indicates from which sessions you can issue them. The mode of the system data mover (SDM) determines which command you can issue.

Table 21. Extended remote copy time sharing option (TSO) commands

TSO command	Can this command be issued on: (1)						
	An active session?	An inactive session?	A suspended session?	A recovery session?	A coupled, active session?	A coupled, inactive session?	A master session?
XADDPAIR	Yes	No	No	No	Yes	No	No
XADVANCE	Yes	Yes	Yes	No	Yes	Yes	No
XCOUPLE ADD	Yes	No	No	No	No	No	No
XCOUPLE DELETE	No	No	No	No	Yes	No	No
XCOUPLE PURGE	No	No	No	No	No	Yes	No
XCOUPLE RELEASE	No	No	No	No	No	No	Yes
XDELPAIR	Yes	No	No	No	Yes	No	No
XEND	Yes	No	No	No	Yes (2)	No	Yes (3)
XQUERY	Yes	No	No	Yes	Yes	No	Yes
XQUERY MASTER	No	No	No	No	No	No	Yes
XRECOVER	No	Yes	Yes	Yes	No	Yes	No
XSET	Yes	No	No	No	Yes	No	No
XSTART	No	Yes	Yes	No	No	Yes	No
XSTATUS	Yes	Yes	Yes	Yes	Yes	Yes	Yes
XSUSPEND (Volume)	Yes	No	No	No	Yes	No	Yes (3)
XSUSPEND (Session)	Yes	No	No	No	Yes (2)	No	Yes (3)

Note:

1. There are several exceptions to the guidelines listed above. See the individual commands in this chapter for the exceptions for each command.
2. You can issue the XEND and the XSUSPEND commands to a coupled, active session after all volumes have been suspended (AVS) or there are no volumes in the session (NOV).
3. The XEND, XSUSPEND (VOLUME) and XSUSPEND (SESSION) commands issued to a master session causes the specified command to be coordinated on the member coupled sessions.
4. The XSTATUS command does not have a session parameter, and hence might be issued on any LPAR.
5. You can not specify a cluster name as the session_id for any of the XRC TSO commands except the following situations:
   - When the PARMLIB parameter is specified in the XSET command.
   - When the ENVIRONMENT(PARM) parameter is specified in the XQUERY command.

For additional information about TSO commands, refer to *z/OS TSO/E Command Reference*.

For additional information about TSO sessions, refer to “XRC session state descriptions” on page 30.

XADDPAIR–Adding volume pairs or utility volumes

You can use the XADDPAIR command for the following functions:

- Define primary and secondary volumes that you want to add to an XRC session. When the command processes, XRC copies all changes from the specified primary volume to the specified secondary volume, unless you specify NOCOPY.
- Add volume pairs that were previously suspended. XRC automatically resynchronizes suspended pairs and processes them ahead of new volume pairs (which require a full-volume copy).
- Add a utility volume to a storage control session.
- Specify fix or float utility device support.

You can specify a maximum of 50 volume pairs with a single XADDPAIR command. The primary and secondary volumes must be compatible. For example, they must meet the following conditions:

- The same number of bytes on each track
- The same number of tracks on each cylinder
- The same or larger number of cylinders on the secondary volume

You can specify a maximum of 16 storage control session IDs per LSS by using the XADDPAIR command. You can also add and remove storage control sessions by adding and removing XRCUTL volumes with the same storage control session ID for the fixed utility. You can use the XADDPAIR command to add utility volumes for the XRC sessions with enhanced multiple reader function.

The XADDPAIR command by default performs an image copy of the primary volume to the secondary volume, unless the volume is a utility volume. Applications might continue to update the primary volumes while the XADDPAIR copy operation is processing.

Note: Multiple-volume data sets require special handling. To ensure that the entire data set is recoverable, add *all volumes* of a multivolume data set to the session.

For additional information about utility devices, refer to “Using XRC utility devices” on page 165.

For additional information about the XADDPAIR command, refer to “Adding an XRC volume pair” on page 170.

XADDPAIR command syntax

```
►►──┬─XADDPAIR─┬──session_id──┬─┤ A: other XADDPAIR options ├─┬─────────────►
    └─XADD─────┘              └─SUSPENDED─────────────────────┘

►──┬──────────────────────────┬──────────────────────────────────────────►◄
   └─MSGROUTEID──(user ID)────┘
```

A: other XADDPAIR options: You can issue any of the following parameters in place of the SUSPENDED parameter.

```
                  ┌──────────────────────────────────────┐
                  ▼                                      │
►►──VOLUME──(───primary_volser── ──┬─secondary_volser─┬──┴──)──┬──────────────────────────────────┬──►
                                   └─XRCUTL───────────┘        │                ┌─ON──┐           │
                                                               ├─DVCBLOCK──(────┼─OFF─┼───)───────┤
                                                               │                └─WPn─┘           │
                                                               └─DONOTBLOCK───────────────────────┘

                                                       ┌─FULLCOPY──┐
►──┬──────────────────────────────────────────────┬────┼───────────┼──┬──────────────────────────────────────────┬──►◄
   └─┬─ERRORLEVEL─┬──(──┬─SESSION────┬──)─────────┘    ├─NOCOPY────┤  ├─SCSESSION──(storage_control_session)───┤
     └─ERRLVL─────┘     ├─VOLUME─────┤                 └─QUICKCOPY─┘  └─LOGPLUS─────────────────────────────────┘
                        └─group_name─┘
```

Required parameters:

session_id
: Specifies the name of an XRC session. The name you specify can be any name acceptable to TSO that matches the name of a session previously specified on the XSTART command, except for the word ALL, which is reserved. The maximum length of the name is eight characters. Do not include embedded blanks. TSO ignores leading blanks and trailing blanks.

VOLUME
: Specifies the volume serial numbers of the primary and secondary volume pairs you want to add to an XRC session. Use either a blank or a comma to separate each primary and secondary volume serial number within each pair and to separate the pairs. The VOLUME parameter is mutually exclusive with the SUSPENDED parameter.

XRCUTL
: Specifies that the system data mover can use the primary volume as a utility volume for a storage control session. This means that a utility volume is *not* shadowed to a secondary volume when the secondary volser specified is XRCUTL.

Specifying XRCUTL in conjunction with the UTILITY parameter on the XSET command determines the type of device (fixed or floating) that XRC uses to read data from a storage control.

When you specify XRCUTL, XRC ignores the NOCOPY, DONOTBLOCK, ERRORLEVEL, FULLCOPY, and QUICKCOPY options.

SUSPENDED
: Specifies that XRC adds all suspended volumes back into the session and schedules them for resynchronization. This parameter is mutually exclusive with the VOLUME, DONOTBLOCK, NOCOPY, SCSESSION, ERRORLEVEL, FULLCOPY, and QUICKCOPY parameters. Suspended volumes keep the same characteristics that existed when the suspension occurred.

Note: Do not use this option to add volumes back into the session if you plan to change any of the volume pair characteristics (such as error level, session assignment, or blocking assignment)

For additional information about XRC utility devices, refer to “Using XRC utility devices” on page 165.

Optional parameters:

DVCBLOCK

Specifies the device blocking option to be applied to the volume or list of volumes in a session specified with the VOLUME parameter. The DVCBLOCK parameter is mutually exclusive with the DONOTBLOCK parameter and is ignored if you have specified SUSPENDED or XRCUTL.

Workload-based write pacing affects the behavior of DVCBLOCK. For more information, refer to “Workload-based write pacing” on page 161.

The DVCBLOCK values are:

ON Specifies that device blocking is enabled for the specified volume(s). This is the default unless the SHADOW DfltWritePacingLvl PARMLIB value is specified as non-zero. When DfltWritePacingLvl is non-zero, all volumes on write pacing capable storage controllers are write paced at the defaulted level unless overwritten.

When workload-based write pacing is in use, XRC automatically converts DVCBLOCK(ON) to DVCBLOCK(WP*n*), where *n* is the discretionary level, 6.

OFF Specifies that device blocking and write pacing are disabled for the specified volume(s).

WP*n* Specifies that write pacing is to be activated for the specified volume(s), with *n* specifying the level of write pacing delay, 0-F.

WP0 specifies that the session default level will be used, as specified the SHADOW DfltWritePacingLvl PARMLIB value.

WP1-WP7 result in pacing maximums of 0.02, 0.04, 0.1, 0.5, 1, and 2 milliseconds per recordset, respectively. These levels are useful for volumes with high rates of small blocksize writes, such as data base logs, where minimal response time impact is essential.

WP8-WPC result in pacing maximums of 5, 10, 25, 50, and 100 milliseconds per recordset, respectively. These levels are useful for volumes with high mb/sec write rates.

WPD-WPF result in pacing maximums of 200, 500, and 1000 milliseconds per recordset, respectively. These levels should be used only in exceptional situations where a very high degree of pacing is required.

Delay is injected per recordset, but a write channel program might create several recordsets. In such cases, the maximum possible delay per write channel program is equal to the pacing maximum multiplied by the number of recordsets that the channel program creates. For example, at pacing level WPC, a channel program that creates 4 recordsets can be delayed for a total of 4×100, or 400 milliseconds.

There is no overall limit on the amount of delay that can be injected for a channel program. Very large channel programs, such as those used in Sort applications, can experience delays measured in seconds when higher pacing levels are used.

DONOTBLOCK

Specifies that write pacing is disabled and that the application I/O to the specified primary volumes is *not* blocked, even if the system data mover is unable to offload updates from the volumes rapidly enough. If you do not specify this parameter, the default is to block the primary volumes, on storage controls that support this function. This parameter is ignored if you have specified SUSPENDED or XRCUTL. You cannot use the DONOTBLOCK parameter with the DVCBLOCK parameter, or with devices on write-pacing-capable controllers.

Workload-based write pacing affects the behavior of DONOTBLOCK. For more information, refer to “Workload-based write pacing” on page 161.

ERRORLEVEL

Specifies how the XRC session responds when an error occurs that causes the session to become unable to process a volume. You can abbreviate this parameter as **ERRLVL.**

Note: Suspended volumes keep the same characteristics that existed when the suspension occurred.

The error level you specify with the XADDPAIR command overrides the value that was specified in the XSTART command. If you do not specify the ERRORLEVEL parameter, the error level defaults to the value that was specified on the XSTART command.

For a volume resynchronization, you can change the error level for a volume pair from what it was prior to the suspension. XRC ignores this parameter for utility volumes or if you also specify SUSPENDED.

To specify how XRC should process certain error conditions, select one of the values:

SESSION

Specifies that, if a permanent error that is associated with a duplex primary or secondary volume occurs, XRC suspends *all volume pairs* in the session, regardless of the volume status. If the session is coupled to other XRC sessions, all volumes in the coupled XRC sessions are also suspended.

If you specify the SESSION parameter, volume pairs that are suspended from the session will negatively affect the usability of the remaining secondary volumes. Specify the SESSION parameter to ensure that all secondary volumes necessary for recovery are consistent up to the time of failure.

Note: To couple an XRC session to a master session, use the ERRORLEVEL(SESSION) parameter to ensure a recoverable consistency across all coupled sessions.

VOLUME

Specifies that, if a permanent error occurs, the XRC session suspends only the volume pair or pairs that are associated with the error. All other volumes continue to process.

group_name

Specifies that, if an error to a duplex primary or secondary volume occurs, XRC suspends all volume pairs that are associated with that group name. The *group_name* can be any name acceptable to TSO. The maximum length of the name is eight characters. Do not include embedded blanks. TSO ignores leading blanks and trailing blanks. Do not use the reserved names VOLUME and SESSION.

Note: The *group_name* parameter is not supported across coupled XRC sessions.

NOCOPY

Specifies that XRC add a volume pair to an XRC session without initially copying the full primary volume. XRC *only* copies updates to the volume after it adds the volume. This assumes that the storage administrator had previously copied the volume. XRC puts the volumes directly into DUPLEX status.

If you do not specify NOCOPY, XRC processes as follows:

- XRC copies the primary volume onto the secondary volume at the same time that additional application updates are made to both copies.
- Volumes are initially in PENDING status and go to COPY status when XRC selects them for synchronization.
- When synchronization is complete, the volumes enter DUPLEX status.

Note: XRC ignores NOCOPY when establishing utility volumes and when resynchronizing a volume. XRC copies data to the secondary volumes during resynchronization.

FULLCOPY

Specifies that XRC make an image copy of *each* track of the primary volume onto the secondary volume at the same time that additional application updates are made to both copies. The volumes are initially in pending status and enter duplex state when the initial volume copy completes. FULLCOPY is the default.

Note: If you do not specify a COPY option using the XADDPAIR command, XRC uses the option that is specified on the XSET COPY command. XRC ignores the FULLCOPY option when establishing utility volumes and when resynchronizing a volume. If you do not specify an XSET COPY option, XRC uses the default of FULLCOPY.

QUICKCOPY

Specifies that XRC make an image copy of *allocated* tracks on the primary volume onto the secondary volume at the same time that additional application updates are made to both copies. XRC reads the VTOC of the primary volume to determine the allocated tracks on the volume. The volumes are initially in PENDING status and enter DUPLEX status when the quick volume copy completes.

XRC ignores the QUICKCOPY option when establishing utility volumes and when resynchronizing a volume.

Note: To ensure data integrity, the initial processing for QUICKCOPY must issue a reserve and then a release for the primary volume during the initial phase of the synchronization process. If access to the primary volume is through a channel extender and the connection fails while XRC has the volume reserved, applications at the primary site will not be able to access the primary volume.

SCSESSION

Specifies that XRC assign the volumes that are associated with this command to the value that you specify with *storage_control_session*. You can assign either one or two alphabetic characters for the *storage_control_session* value. If you do not specify this parameter, the default value is “--”.

XRC ignores any change to the SCSESSION designation if this XADDPAIR command is resynchronizing volume pairs or if you also specify the SUSPENDED parameter. If you want to change the volume assignment, you must first issue an XDELPAIR command, then reissue the XADDPAIR command for the volume pair. This will require that you perform a full synchronization copy for the volume pair.

The SCSESSION parameter is valid for utility volumes as well as for regular volume pairs.

LOGPLUS

Specifies that the primary volume of the volume pair is to be explicitly written to by the z/OS System Logger. When using LOGPLUS, you should specify a single pair and a utility pair, but the utility pair must be specified last. A unique storage control session number is assigned to the primary volume. If you do not specify a utility pair, any attempt to reserve the LOGPLUS volume (such as for allocating data sets) will result in a *waiting for control unit* condition.

LOGPLUS cannot be specified with the SCSESSION parameter. For more information, refer to Plan DRXRC-type Staging Data Sets for Coupling Facility Log Streams in *z/OS MVS Setting Up a Sysplex*.

Note: Overloading an individual XRC session with LOGPLUS volumes can result in degraded session performance and unacceptable application system impact. When adding high activity storage control sessions (such as those associated with LOGPLUS volumes), be sure to balance them across XRC sessions, maintaining a ratio of no more than one high activity session for every three lower activity ones within a given session.

MSGROUTEID

Specifies the user ID to which XRC messages associated with this command are routed. If the specified user ID is not logged on, TSO saves messages in the TSO BRODCAST data set and displays them after the next logon.

Note: Messages issued by the initial command syntax-checking routine are sent to the user ID that issues the TSO command.

For additional information about:

- Using the NOCOPY option, refer to “Adding a volume with the NOCOPY option” on page 173.
- Using the QUICKCOPY option, refer to “Adding a volume with the FULLCOPY or QUICKCOPY option” on page 172.
- Using XRC utility devices, refer to “Using XRC utility devices” on page 165.

XADDPAIR command examples

The following are examples of the XADDPAIR command:

```
XADDPAIR MAIN1 VOLUME(PRIM02 SECD02) ERRLVL(VOLUME)

XADDPAIR MAIN1 SUSPENDED

XADDPAIR MAIN1 VOLUME(PRIM01 SECD01) QUICKCOPY
```

For examples of the XADDPAIR command to add utility volumes, refer to “Using XRC utility devices” on page 165.

XADVANCE—Updating secondary volumes

You can issue the XADVANCE command to both coupled and noncoupled sessions.

The XADVANCE command differs from the XRECOVER command in that it obtains a consistent set of secondary volumes without updating the secondary volume VOLSERs. This makes it easier for you to create a time-consistent backup of the secondary volumes before restarting XRC and resynchronizing volumes.

The XADVANCE command requires all volumes in the session to be suspended. It may be used with an active or an inactive session. You cannot use this command when the session is active on another system, or if the session has any nonsuspended volumes. The XADVANCE command does not apply any data to volumes that were suspended because of a command, or that were suspended because of errors that occurred before the last consistency group.

If an error occurs while writing to a secondary volume that results in all volumes being suspended, that volume is suspended at a different consistency time than the other volumes in the session. To resolve the time discrepancy, correct the error on the secondary volume, and issue the XADVANCE command to update that volume to the same consistency time as other volumes in the session.

Use the XADVANCE TSO command to update secondary volumes of a session to a time that is consistent with the volumes of other sessions that are coupled to the master session. The XADVANCE command applies secondary volume updates to the minimum master journal time or the pending master command time. This is the same time to which an XRECOVER command would recover the session.

If RACF facility class protection is being used for XRC, the XADVANCE command should be added to the resource profile using the procedure documented for existing XRC commands. The XADVANCE command must also be defined as an authorized TSO command.

The XADVANCE command produces an XQUERY report that displays *YES* in the ADV column to indicate each volume in the session that was eligible to have updates applied. A *NO* in the column indicates that the volume was not eligible to have updates applied. The report also displays *NO* for volumes that have not reached duplex, that were suspended through a command, or that were suspended at a time before the consistency time that exists as of the beginning of the XADVANCE operation.

Issuing an XADVANCE command to a master session automatically generates an XQUERY MASTER report for the coupled sessions.

See “Example: resuming operations after a sympathetic suspension” on page 230 for a scenario that uses XADVANCE to resume XRC operations after a sympathetic suspension caused by an error.

XADVANCE command syntax

The syntax of the XADVANCE command is:

```
►►──XADVANCE──session_id──┬─────────────────────────┬──┬───────────────────────────┬──────────►
                          │           ┌─SYS1─┐      │  │             ┌─SYS1──┐     │
                          └─HLQ──(────┴─hlq──┴───)──┘  └─MHLQ──(─────┴─mhlq──┴───)─┘
```

```
►─┬────────┬─┬──────────────────────────┬────────────────────────►◄
  └─ONLINE─┘ └─MSGROUTEID──(──user_ID──)─┘
```

Required parameters:

session_id
: Specifies the name of an XRC session that is to have its secondary volumes updated to a consistent time. This can be any name acceptable to TSO that matches the name of a session that was previously specified on the XSTART command. The maximum length of the name is eight characters. Do not include embedded blanks. TSO ignores leading blanks and trailing blanks.

Optional parameters:

HLQ_(hlq)_
: Specifies the name of the high-level qualifier for the state, control, and journal data sets which was previously specified on the XSTART command. The HLQ name must be from one to eight characters, and can be any name acceptable to TSO. SYS1 is the default value.

MHLQ_(mhlq)_
: Specifies the name of the high-level qualifier for the master session control data set, which was previously specified on the XCOUPLE command. The MHLQ name must be from one to eight characters, and can be any name acceptable to TSO. SYS1 is the default value.

ONLINE
: Specifies that any online volume that matches the volume serial number (*volser*) of a secondary volume will be used as a secondary volume when this command processes.

MSGROUTEID
: Specifies the user ID to which XRC messages associated with the processing of this command are routed. If the specified user ID is not logged on, TSO saves messages in the TSO BRODCAST data set and displays them after the next logon.

  Note: Messages issued by the initial command parser routine are sent to the user ID that issues the TSO command.

For additional information about secondary volumes, refer to "Accessing secondary volumes while XRC is active" on page 192.

XADVANCE command example

The following is an example of the XADVANCE command:

```
XADVANCE UMC HLQ(SYS1) MSGROUTEID(OPER3)
```

XCOUPLE—Coupling XRC sessions

Use the XCOUPLE TSO command to add XRC sessions to, or delete XRC sessions from, a master session. You can also use this command to manage the status of the master session. Specifically:

- The XCOUPLE(ADD) command couples the specified XRC session into a master session, which is then used to coordinate the consistency time and data recovery of multiple XRC sessions. When two or more XRC sessions are coupled, the journal and consistency times of the sessions are maintained such that the entire group of sessions can always be processed by the XRECOVER or XADVANCE commands to a consistent point in time. The XCOUPLE(ADD) command is not allowed for sessions where you specified SESSIONTYPE(MIGRATE) on the XSTART command.

  Note: Starting a cluster session in the LPAR that runs XRC, the XCOUPLE(ADD) command couples the XRC session to the specified master session through the cluster session. To couple the XRC session directly to the master session without using a cluster session, disable the cluster session before you issue the XCOUPLE(ADD) command. If a cluster session is currently active in the LPAR and you restart the XRC session, the system recouples an XRC suspended session to the master session.

- The XCOUPLE(DELETE) command uncouples the specified active XRC session from the master session. In a coupled environment, when the last volume is deleted, the system will automatically perform the XCOUPLE DELETE command to remove the session from the master session.
- The XCOUPLE(PURGE) command is issued to an inactive coupled session. If the master session is active, this command clears information about the coupled XRC session from the master data set.
- The XCOUPLE(RELEASE) command removes the HOLD status of the master session. Use the RELEASE option only if you no longer need to protect the recoverable state of the coupled XRC sessions in the master session and are ready to resume normal remote copy operations.

Note: If you issue the XSTART(*session_id*) command to an XRC session using SESSIONTYPE(MIGRATE), you cannot issue the XCOUPLE command to couple the session to a master session. If the XRC session has been migrated and you want to couple it, follow these steps:

1. Issue the XEND or XSUSPEND command to the session.
2. Restart the session by using SESSIONTYPE(XRC).
3. Issue the XCOUPLE command to the XRC session

If you restart a coupled session and specify SESSIONTYPE(MIGRATE), the session type is changed to XRC. To restart a coupled session using MIGRATE, you must first issue the XCOUPLE PURGE command to uncouple the session.

Note:

1. If RACF facility class protection is being used for XRC, the XCOUPLE command should be added to the resource profile by using the procedure documented for existing XRC commands. The XCOUPLE command must also be defined as an authorized TSO command.
2. The maximum number of coupled sessions that are supported by this release of CXRC is 14.

If you issue an XSTART command to restart a previously coupled session, and the automatic XCOUPLE ADD processing fails, subsequent XADDPAIR commands also fail with a return code indicating that the XCOUPLE processing is incomplete. To resolve this condition and to permit XADDPAIR commands to complete, you must either:

- Correct the condition that caused the XCOUPLE ADD command to fail and reissue the XCOUPLE ADD command to the XRC session that encountered the problem.
- Issue the XCOUPLE DELETE command to uncouple the XRC session.

For additional information about coupled XRC sessions, refer to "Managing coupled XRC operations" on page 219.

XCOUPLE command syntax

The XCOUPLE command includes two syntax diagrams:

- The first syntax diagram allows you to add, delete, or purge as follows:
  - Add (*couple*) the specified XRC session to the group of sessions that is associated with the master session.
  - Delete (*uncouple*) the specified XRC session from the group of sessions that is associated with the master session.
  - Purge information about the specified coupled XRC session from the master data set when the XRC session is inactive.
- The second syntax diagram allows you to remove the HOLD status of the master session.

```
►►──XCOUPLE──session_id──┬─ADD───┬─MSESSION──(──msession_id──)──┬───────┬──────────►
                         │       │           ┌─SYS1─┐           │       │
                         │       └─MHLQ──(───┴─mhlq─┴───)───────┘       │
                         ├─DELETE──┬──────────┬─────────────────────────┤
                         │         └─NOVERIFY─┘                         │
                         └─PURGE───┬─MSESSION──(──msession_id──)──┬─────┘
                                   │          ┌─SYS1─┐            │
                                   ├─HLQ──(───┴─hlq──┴───)────────┤
                                   │           ┌─SYS1─┐           │
                                   └─MHLQ──(───┴─mhlq─┴───)───────┘

►──┬────────────────────────────────┬─────────────────────────────────────────────►◄
   └─MSGROUTEID──(──user ID──)──────┘
```

Required parameters:

session_id
: Specifies the name of an XRC session that is to have its coupling status changed. It can refer to an uncoupled session or a coupled XRC session. This name can be any name acceptable to TSO that matches the name of a session that was previously specified on the XSTART command. The maximum length of the name is eight characters. Do not include embedded blanks. TSO ignores the leading blanks and the trailing blanks.

(ADD | DELETE | PURGE)
: **(ADD)** indicates that the specified XRC session *session_id* is to be added or coupled to the group of sessions that is associated with the master session. **(DELETE)** indicates that the specified active XRC session *session_id* is to be removed or uncoupled from the group of sessions associated with the master session. **(PURGE)** removes information about the XRC session from the master data set when the XRC session is inactive.

MSESSION(*msession_id)*
: Specifies the name of the master session. The master session is a logical entity that is used to coordinate session commands and data consistency across multiple XRC sessions. A master session exists as long as there is an XRC session coupled to the master session. The maximum length of the name is eight characters. Do not include embedded blanks. TSO ignores the leading blanks and the trailing blanks. This parameter is not used on the XCOUPLE DELETE command.

Optional parameters:

HLQ*(hlq)*
: Specifies the name of the high-level qualifier for the state, control, and journal data sets to which the XCOUPLE command refers. The XCOUPLE HLQ name must be from one to eight characters, and can be any name acceptable to TSO. SYS1 is the default value. HLQ is only valid with the XCOUPLE PURGE command.

MHLQ*(mhlq)*
: Specifies the high-level qualifier of the master session control data set. The default is SYS1. MHLQ is only valid with the XCOUPLE ADD and XCOUPLE PURGE commands. It is not a valid option for the XCOUPLE DELETE command.

MSGROUTEID
: Specifies the user ID to which XRC messages that are associated with the processing of this command are routed. If the specified user ID is not logged on, TSO saves messages in the TSO BRODCAST data set and displays them after the next logon.

  Note: Messages issued by the initial command parser routine are sent to the user ID that issues the TSO command.

NOVERIFY
: Specifies that XRC not issue message ANTT0025A, which prompts the user to confirm that XRC should continue to process the command. If you select this option, XRC immediately processes the command without waiting for verification.

Remove the HOLD status of the master session: The syntax of the XCOUPLE command to remove the HOLD status of the master session is:

```
►►──XCOUPLE──msession_id──RELEASE──┬──────────────────────────────────────────┬──────►
                                   │            ┌─SYS1─┐                      │
                                   └─MHLQ──(──┴─mhlq─┴──)──┬────────────┬─┘
                                                           └─NOVERIFY─┘

►──┬────────────────────────────────┬───────────────────────────────────────────►◄
   └─MSGROUTEID──(──user ID──)─┘
```

Required parameters:

msession_id
: Specifies the name of the master session. The master session is a logical entity that is used to coordinate session commands and data consistency across multiple XRC sessions. A master session exists as long as there is an XRC session coupled to the master session. The maximum length of the name is eight characters. Do not include embedded blanks. TSO ignores the leading blanks and the trailing blanks.

RELEASE
: Indicates that all manual, operator-initiated actions have been performed to protect the recoverable state of coupled XRC sessions that are associated with the specified master session. The HOLD status of the master session is to be removed.

Optional parameters:

MHLQ*(mhlq)*
Specifies the high-level qualifier of the master session control data set. The default is SYS1.

MSGROUTEID
Specifies the user ID to which XRC messages associated with the processing of this command are routed. If the specified user ID is not logged on, TSO saves messages in the TSO BRODCAST data set and displays them after the next logon.

Note: Messages issued by the initial command parser routine are sent to the user ID that issues the TSO command.

NOVERIFY
Specifies that XRC not issue message ANTT0025A, which prompts the user to confirm that XRC should continue to process the command. If you select this option, XRC immediately processes the command without waiting for verification.

XCOUPLE command examples

The following are examples of the XCOUPLE command:

```
XCOUPLE STJOSEPH ADD MSESSION(HOSPITAL) MHLQ(MSTRHOSP)

XCOUPLE GENERAL DELETE MSGROUTEID(OPER2)

XCOUPLE STJOSEPH PURGE MSESSION(HOSPITAL) MHLQ(MSTRHOSP) HLQ(STJOS1)

XCOUPLE HOSPITAL RELEASE MHLQ(MSTRHOSP)
```

XDELPAIR–Deleting volume pairs

Use the XDELPAIR command to specify that XRC delete volume pairs from an XRC session. When the XDELPAIR command processes, XRC stops processing either a primary volume and its associated secondary volume, or a utility volume. You can delete up to 100 volume pairs by listing the 100 primary volume serial numbers in a single XDELPAIR command.

For each volume, the XDELPAIR command returns a timestamp indicating that the data on the secondary volume is consistent with the data that was on the primary volume at the specified time.

If you issue an XDELPAIR command for one or more volume pairs in a session, the command operates the same in either a coupled or non-coupled environment. Deleting the last volume in a coupled session will cause the session to be uncoupled from the master data set.

If you issue an XDELPAIR command to delete a volume in an enhanced session, the volume is removed from the primary and all auxiliary sessions.

If you issue an XDELPAIR command to delete a volume that is an XRCUTL utility device in an enhanced session, and the parmlib value for number of sessions is not specified, then delete the utility device to decrease the number of auxiliary sessions. To reduce the potential impact of the decrease, suspend all volumes in the enhanced session. If the volumes are suspended, the following additional processing occurs:

- All volumes are withdrawn from one of the auxiliary sessions.

- The auxiliary session is drained of all remaining updates in cache.
- The auxiliary session is terminated.
- The specified XRCUTL device is removed from all remaining primary and auxiliary sessions.
- Remaining utility devices are reassigned within the remaining enhanced session members.

If you issue an XDELPAIR command to delete a utility volume in an enhanced session, but the operation reduces the number of volumes available to read below the NumberReaderTasks value , the XDELPAIR command is rejected.

Note:

1. XRC does not process any volumes in a list if there is a command syntax error, or if any of the volumes in the list are not part of the XRC session.
2. You can delete XRC volume pair with the XDELPAIR command, regardless of the status of the pair. Volume pairs in pending, duplex, copy, seqcheck, and suspend status all become inactive when deleted, and are no longer eligible for recovery.
3. An XRC session remains active even when you have deleted all of its pairs. Issue an XEND command to end the session or an XSUSPEND command to suspend the session.
4. Large batches of XDELPAIR commands issued to Duplex volumes may result in significantly increased session delays. To avoid these delays, consider issuing the XDELPAIR commands while all volumes are suspended, or issuing them in small batches across a period of several minutes.
5. To delete a volume pair that was added in the same request with a utility pair using the LOGPLUS parameter, specify only the primary volume. The utility volume pair is deleted automatically and should not be specified in the XDELPAIR command.

For additional information about deleting utility volumes, refer to “Using XRC utility devices” on page 165.

XDELPAIR command syntax

```
>>──┬─XDELPAIR─┬──session_id──────────────────────────────────────────►
    └─XDEL─────┘

              ┌──────────────────┐
              ▼                  │     ┌─IMMEDIATE───────────────────────────────┐
►─VOLUME──(────primary_volser────┴──)──┼─ATTIME──(──yyyy.ddd hh:mm:ss.thmiju──)─┼──────►
                                       ├─DRAIN───────────────────────────────────┤
                                       └─CANCEL──────────────────────────────────┘

►──┬──────────┬──┬─────────────────────────┬──────────────────────────────────►◄
   └─NOVERIFY─┘  └─MSGROUTEID──(user ID)───┘
```

Required parameters:

session_id
: Specifies the name of an XRC session. This can be any name acceptable to TSO that matches the name of a session that was previously specified on

the XSTART command. The maximum length of the name is eight characters. Do not include embedded blanks. TSO ignores leading blanks and trailing blanks.

VOLUME(*primary_volser*)
Specifies the volume serial numbers of the primary volumes to be deleted from an XRC session. To specify multiple volume serial numbers, separate the volume serial numbers with a blank or a comma. Do not specify the volume serial numbers of the secondary volumes. The XDELPAIR command automatically removes the associated secondary volumes from the session. You can specify up to 100 volumes.

See the IMMEDIATE, ATTIME, and DRAIN parameters for information that describes when volumes are deleted based on the value specified with the VOLUME parameter.

For additional information the selection of the utility device, refer to "Addressing the utility device selection" on page 165.

Optional parameters:

ATTIME
Specifies that XRC delete the specified volume pairs after applying all updates, up to the specified time, to the secondary volumes.

XRC will not apply any updates beyond the specified Universal Time Coordinated (UTC) timestamp to the secondary volumes. All affected secondary volumes are consistent up to the UTC timestamp that is reported on successful completion of the command.

You can issue another XDELPAIR command to change the ATTIME value to either earlier than or later than a previously specified time. The command fails if the specified ATTIME value is earlier than the consistency group time that XRC is currently processing. If you issue multiple commands to a single volume pair, XRC always uses the last valid ATTIME value that you specified with this command.

Note: You cannot issue an XDELPAIR command with the ATTIME parameter to delete a volume that is in the suspended state. You can only delete suspended volumes with an XDELPAIR IMMEDIATE command or an XEND command.

If all volume pairs are deleted or suspended (because of an error or by command) before the time specified on the ATTIME expires, XRC cancels the XDELPAIR command and clears the ATTIME timestamp.

The system data mover (SDM) does not process record updates for the deleted volume that are received in the storage control after the specified time. Record updates in the journal data set that are associated with the deleted volume are discarded and are not available for a subsequent recovery. The storage control continues to record updates to other primary volumes. The SDM then processes these updates.

Specifying the ATTIME parameter with VOLUME(*primary_volser*) results in the following:

- If the specified volser is a fixed utility device, the command is rejected, with return code 4082.
- If the specified volser is not a fixed utility device, it is deleted based on the time specified on the ATTIME parameter.

If more than one volser is specified, and one of the volsers is for a fixed utility device, the command is rejected. If the list of volsers does not include a fixed utility device, XRC will delete each volume in the list based on the time specified on the ATTIME parameter. (The list of volsers can be across multiple storage controls sessions.)

CANCEL
Specifies that XRC cancel the previous XDELPAIR command with an ATTIME or DRAIN request. The CANCEL option is only valid when there is a pending XDELPAIR command.

DRAIN
Specifies that XRC:

1. Reads the record updates from the cache on all storage controls and obtains the timestamp of the last update for each storage control session.
2. Determines the most recent timestamp.
3. Processes the XDELPAIR command when all records, up to and including the most recent timestamp, have been applied to the secondary volumes that are being deleted from the session.

Note: You cannot issue an XDELPAIR command with the DRAIN parameter to delete a volume that is in the suspended state. You can only delete suspended volumes with an XDELPAIR IMMEDIATE command or an XEND command.

If all volume pairs are deleted or suspended (because of an error or by command) before XRC can drain all updates from the primary storage control, XRC cancels the XDELPAIR(DRAIN) command.

Specifying the DRAIN parameter with VOLUME(*primary_volser*) results in the following:

- If the specified volser is a fixed utility device, the command is rejected, with return code 4082.
- If the specified volser is not a fixed utility device, it is deleted after all updates from the storage control cache have been applied to the secondary volumes.

If more than one volser is specified, and one of the volsers is for a fixed utility device, the command is rejected. If the list of volsers does not include a fixed utility device, XRC will delete each volume in the list after all updates from the storage control cache have been applied to the secondary volumes. (The list of volsers can be across multiple storage controls sessions.)

IMMEDIATE
Specifies that XRC immediately process the delete request. IMMEDIATE is the default.

XRC deletes the volume pairs when the current consistency group has been applied. The storage control continues to record updates to other primary volumes, and the SDM processes these updates. The secondary volumes are consistent up to the UTC timestamp that is reported on the successful completion of the command.

Specifying the IMMEDIATE parameter with VOLUME(*primary_volser*) results in the following:

- If the specified volser is not a fixed utility device, it is deleted.

- If the specified volser is the only device in the storage control session and it is a fixed utility device, the volume is deleted. Otherwise, the command is rejected, with return code 4082.

If more than one volser is specified, and one of the volsers is for a fixed utility device and it is not the only device in the storage control session, the command is rejected. If the list of volsers does not include a fixed utility device, XRC will delete each volume in the list. (The list of volsers can be across multiple storage controls sessions.)

NOVERIFY
Specifies that XRC not issue message ANTT0025A, which prompts the user to confirm that XRC should continue to process the command. If you select this option, XRC immediately processes the command without waiting for verification.

MSGROUTEID
Specifies the user ID to which XRC messages associated with the processing of this command are routed. If the specified user ID is not logged on, TSO saves messages in the TSO BRODCAST data set and displays them after the next logon.

Note: Messages issued by the initial command parser routine are sent to the user ID that issues the TSO command.

XDELPAIR command examples

The following are examples of the XDELPAIR command:

```
XDELPAIR MAIN1 VOLUME(PRIM01 PRIM04 PRIM05) ATTIME(1997.234 12:14:20.002001)

XDELPAIR MAIN1 VOLUME(PRIM02 PRIM03) DRAIN
```

XEND–Ending a session

Use the XEND command to stop all XRC activity to active volumes and to end the XRC session. Issuing the XEND command stops all updates to the secondary volumes at the time specified.

In a coupled environment, you can issue the XEND command to a master session to have all sessions end with their volumes consistent to the same time.

The XEND command returns a timestamp indicating that the data on the secondary volumes of all active pairs is consistent with the corresponding data that was on the primary volumes at the specified time.

After you issue the XEND command, XRC prompts you (unless you have specified NOVERIFY) to confirm the request to end the session. You must specify either `YES` or `GR215NO`. If the XEND command is invoked through a CLIST written in REXX, you should either specify NOVERIFY or place your response to the prompt in the data stack before issuing the command. In a batch environment, you can stack the response.

An XEND command that specifies an ATTIME value overrides a time that was specified with a previous XEND command. Also, an XEND command that specifies an ATTIME value overrides a pending XSUSPEND(*session_id*) command function.

All secondary volumes are consistent up to the UTC timestamp that is reported by successful completion of the command. The storage control session ends after the time that is determined by the XEND command. Record updates in the journal data set are discarded and not available for a subsequent recovery. If you issue an XRECOVER command, the secondary volume serial numbers of volume pairs in the duplex or seqcheck state are changed to match (“clipped to”) the primary volume serial number, but no journal updates are applied to the volumes. The secondary volume serial numbers of suspended volume pairs are also clipped, provided that the pair had completed initial volume synchronization prior to being suspended.

For information about how to end coupled sessions, refer to “XEND in a coupled environment” on page 89.

XEND command syntax

The syntax of the XEND command is:

```
                        ┌─IMMEDIATE──────────────────────────────┐
►►──XEND──session_id────┼────────────────────────────────────────┼──┬───────────┬──►
                        ├─ATTIME──(──yyyy.ddd hh:mm:ss.thmiju──)─┤  └─NOVERIFY──┘
                        ├─DRAIN──────────────────────────────────┤
                        └─CANCEL─────────────────────────────────┘

►──┬──────────────────────────────┬──────────────────────────────────────────────►◄
   └─MSGROUTEID──(user ID)────────┘
```

Required parameters:

session_id
: Specifies the name of an XRC session. This can be any name acceptable to TSO that matches the name of a session previously specified on the XSTART command. The maximum length of the name is eight characters. Do not include embedded blanks. TSO ignores leading blanks and trailing blanks.

Optional parameters:

IMMEDIATE
: Specifies that XRC immediately ends the active session. XRC stops updates to the secondary volumes of active pairs once the current consistency group has been applied. Secondary volumes are consistent up to the UTC timestamp that is reported on the successful completion of the command. IMMEDIATE is the default.

ATTIME
: Specifies that XRC end the specified XRC session after applying all updates, up to the time that is specified, to the secondary volumes. XRC will not apply updates to the secondary volumes beyond the specified universal time coordinated (UTC) timestamp.

When XRC encounters a record update that has a timestamp later than the time specified, XRC ends the session. All affected secondary volumes are consistent up to the UTC timestamp that is reported on successful completion of the command.

You can issue another XEND command to change the ATTIME value to either earlier than or later than a previously specified time. XRC issues a message that indicates when it has accepted a new ATTIME value. The command fails if the specified ATTIME value is earlier than the consistency group time that XRC is currently processing. If you issue multiple XEND ATTIME commands, XRC always uses the last valid ATTIME value that is specified.

If all volume pairs become deleted or suspended (either due to an error, or by command) prior to the specified ATTIME, XRC cancels the XEND command and clears the ATTIME timestamp.

DRAIN

Specifies that XRC:

1. Read the record updates from all storage control caches and obtain the timestamp of the last update for each storage control session.
2. Determine the most recent timestamp.
3. Perform the XEND command when XRC has applied all records, up to and including the most recent timestamp, to the secondary volumes.

The DRAIN parameter supports operations such as migration, where applications are quiesced prior to issuing the DRAIN request. This parameter allows you to quiesce application program updates to the primary volumes and ensures that all updates have been applied to the secondary volumes before the session is ended.

If non-timestamped record updates exist in a storage control, the DRAIN function ensures that all updates in the storage control at the time the command was issued have been successfully applied to the secondary volume or volumes before the command is performed.

If all volume pairs become deleted or suspended (either due to an error, or by command) before XRC can drain all updates from the primary storage control, XRC cancels the XEND(DRAIN) command.

CANCEL

Specifies that XRC cancel the previous XEND command that was issued with an ATTIME or DRAIN request specified. If a pending XEND command was originally issued using the name of the master session, you cannot issue the CANCEL option for that command using the name for the local session.

NOVERIFY

Specifies that XRC not issue message ANTT0022A, which prompts the user to confirm that XRC should continue to process the command. If you select this option, XRC immediately processes the command without waiting for verification.

MSGROUTEID

Specifies the user ID to which XRC messages associated with the processing of this command are routed. If the specified user ID is logged off, TSO saves messages in the TSO BRODCAST data set and displays them after the next logon.

Note: Messages issued by the initial command parser routine are sent to the user ID that issues the TSO command.

XEND command examples

The following are examples of the XEND command:

```
XEND MAIN1 MSGROUTEID(OPER1) DRAIN

XEND MAIN1
```

XEND in a coupled environment

In a coupled environment, you can issue the XEND command to a master session to have all sessions end with their active duplex volumes at a consistent time across the coupled sessions. You can issue the XEND command to an individual XRC session only:

- After the session has been uncoupled
- If the session is in a coupled state with all volumes suspended
- After there are no volumes in the session

If you uncouple an XRC session using the XCOUPLE command with the DELETE option before you issue the XEND command, the session will no longer be consistent with the other sessions in the master session. Therefore, you will not be able to recover all volumes to a consistent state with the other volumes in the master session.

You may end an XRC session that is coupled in one of three ways:

- Issue an XEND to the master session.
- Suspend or delete all volumes in the XRC session and then issue an XEND to the XRC session.
- Issue an XCOUPLE DELETE to the XRC session and then issue an XEND to the XRC session.

XEND processing for a coupled session will automatically generate an XQUERY MASTER report.

For additional information about the XQUERY MASTER report, refer to “XQUERY MASTER in a coupled environment” on page 98.

XEND in a coupled environment command syntax: The syntax of the XEND in a coupled environment command is:

```
►►──XEND──msession_id──┬─────────────────────────────┬──────────────────────►
                       │          ┌─SYS1─┐           │
                       └─MHLQ──(──┴─mhlq─┴──)────────┘

   ┌─IMMEDIATE──────────────────────────────────┐
►──┼────────────────────────────────────────────┼──┬──────────┬─────────────►
   ├─ATTIME──(──yyyy.ddd hh:mm:ss.thmiju──)─────┤  └─NOVERIFY─┘
   ├─DRAIN──────────────────────────────────────┤
   └─CANCEL─────────────────────────────────────┘

►──┬──────────────────────────┬─────────────────────────────────────────────►◄
   └─MSGROUTEID──(user_ID)────┘
```

Required parameters:

msession_id

Specifies the name of the master session. The master session is a logical entity that is used to coordinate session commands and data consistency across multiple XRC sessions. A master session exists as long as there is an XRC session coupled to the master session. The maximum length of the name is eight characters. Do not include embedded blanks. TSO ignores the leading blanks and the trailing blanks.

Optional parameters:

MHLQ*(mhlq)*
: Specifies the high-level qualifier of the master session control data set. The default is SYS1.

IMMEDIATE
: When you issue the XEND command by using a master session name, the command is processed on all XRC sessions that are coupled to the master session. In this instance, the IMMEDIATE parameter is processed as ATTIME equal to a time that will permit all coupled sessions to be ended at the same consistency time.

ATTIME
: When the XEND command is issued using a master session name, the ATTIME parameter is processed as global ATTIME if no component session has received an application update timestamp past the requested timestamp. If any session has passed the time, the command is rejected.

DRAIN
: Specifies that all sessions in a coupled master session determine a 'drain time'. When you specify a XEND command by using a master session name, the DRAIN parameter is processed as ATTIME by using the latest component drain time.

CANCEL
: Specifies that XRC cancel the previous master XEND command that has not yet processed. The CANCEL option cancels a pending XEND command that was issued to the master session but not yet processed. If the XEND command has started processing on any session, the CANCEL request fails. (The CANCEL option is valid only when there is a pending XEND command.)

NOVERIFY
: Specifies that XRC not issue message ANTT0022A, which prompts the user to confirm that XRC should continue to process the command. If you select this option, XRC immediately processes the command to end the master session without waiting for verification.

MSGROUTEID
: Specifies the user ID to which XRC messages that are associated with the processing of this command are routed. If the specified user ID is logged off, TSO saves messages in the TSO BRODCAST data set and displays them after the next logon.

  Note: Messages issued by the initial command parser routine are sent to the user ID that issues the TSO command.

XEND in a coupled environment command examples: The following are examples of the XEND in a coupled environment command:

```
XEND MSTR1 MSGROUTEID(OPER1) DRAIN

XEND MSTR3 MHLQ(MSTRMHLQ) IMMEDIATE
```

XQUERY–Querying a session

Use the XQUERY command to request status information that is related to an active XRC session. You can customize your information request using the XQUERY command. An enhancement to the XQUERY function allows you to issue an XQUERY MASTER command to a master session; for clusters in a master session, status information is returned for all XRC sessions in the cluster.

You can generate various summary reports by issuing the XQUERY command with the session ID.

The session consistency group time reported in the XQUERY report is the timestamp of the last application update that was applied to all active secondary volumes in the XRC session. You can use the reported consistency group time to monitor application update activity.

The time reported with IDLE indicates how long the session has been idle. In the IDLE state, XRC has applied all updates to the secondary volumes, and the primary storage controls do not have any updates pending. The time reported with DELAY indicates how far the secondary system lags behind the primary system before all current updates are applied.

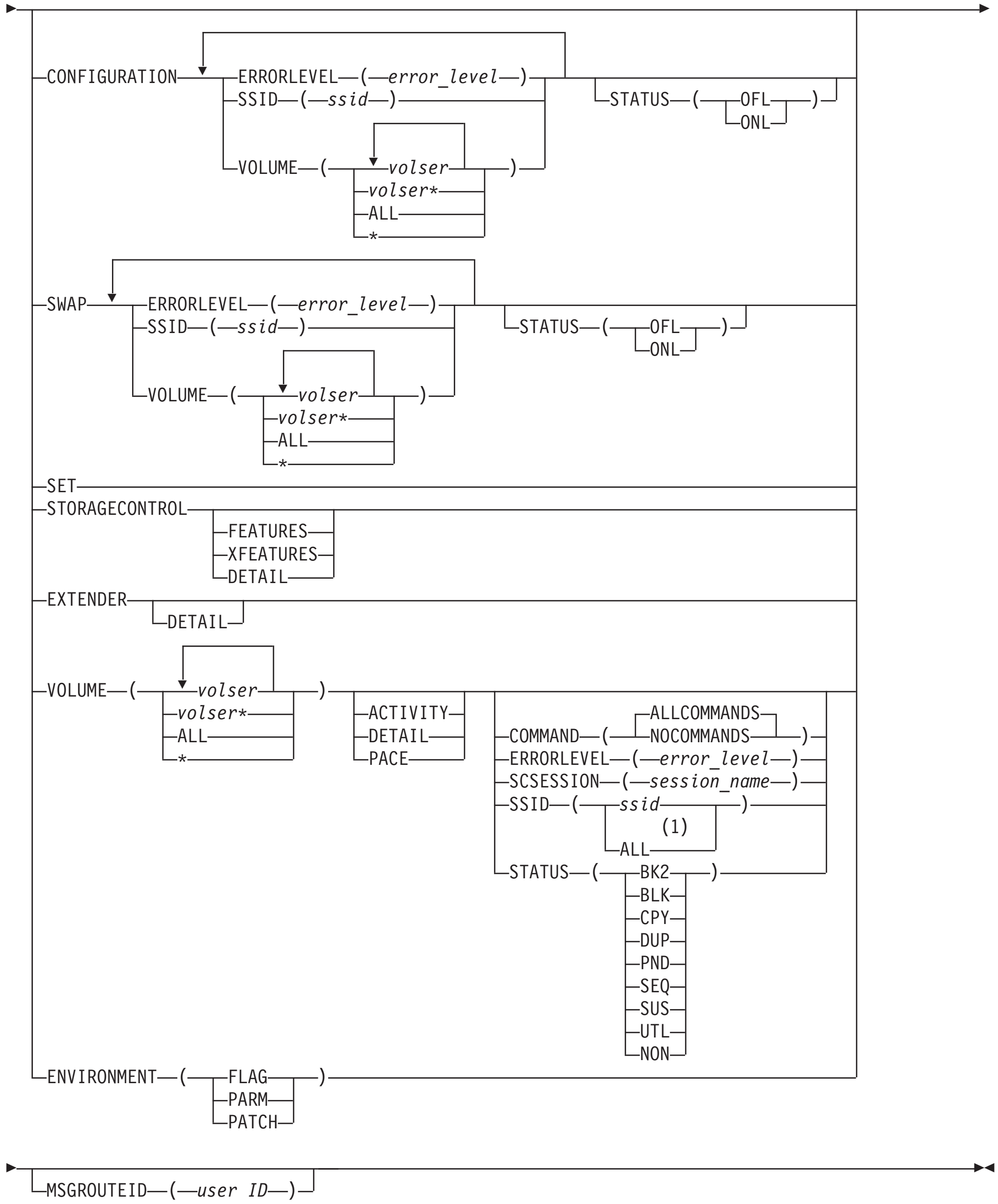
The data exposure time that is reported in the XQUERY summary report provides an approximation of the time difference between data that is written to the primary volumes and data that is secured on the journal data set. If the data exposure time is zero, all data written to primary volumes for the session has been secured on the journal data set. If the data exposure time is greater than zero, then the data that has not yet been secured on the journal data set is potentially nonrecoverable. You might use the data exposure time to identify bottlenecks that might prevent the maximum amount of data from being recovered in the event of a disaster.

For additional information about master session reports, refer to "XQUERY MASTER in a coupled environment" on page 98.

For additional information about the types of reports that can be generated, refer to "Using XRC reports" on page 184.

XQUERY command syntax

```
►►──XQUERY──session_id──┬─────────────────────────────────────────────────────────┬──────────►◄
                        └─DATASET──(──dataset_name──)──┬──────────────────────────┬─┘
                                                       │               ┌─OLD─┐    │
                                                       └─DISP──(───────┼─MOD─┼──)─┘
                                                                       └─SHR─┘
```



Notes:

1 SSID(ALL) is valid only if you specify the VOLUME and DETAIL parameters. Normally, the VOLUME DETAIL output is sorted by the residual count (RES CNT) in descending order, and then by primary volumes (PRIM VOL) in ascending order. If SSID(ALL) is specified, the output is sorted by SSID, then RES CNT, and then PRIM VOL.

Required parameters:

session_id

Specifies the name of an XRC session. This can be any name acceptable to TSO that matches the name of a session previously specified on the XSTART command, or you may specify ALL, which returns information for any active XRC session on this MVS host. The maximum length of the name is eight characters. Do not include embedded blanks. TSO ignores leading blanks and trailing blanks.

If you specify the session ID only, without the SET, CONFIGURATION, STORAGECONTROL, ENVIRONMENT, or VOLUME parameters, XRC generates a summary report of the session status.

Note:

1. When the ENVIRONMENT(PARM) parameter is specified in the XQUERY command, the name of a cluster can be specified as the session name. For all other parameters, you cannot specify a cluster name as the session name.
2. When the ENVIRONMENT(PARM) parameter is specified with a session_id that represents an inactive or undefined session, values for the control address space (ANTAS000) are reported. These values are the default environment values and any overrides from the ANTXIN00 parmlib member.

Optional parameters:

DATASET
: Specifies that XRC direct the XQUERY output to the sequential data set that is identified by *dataset_name*. Standard TSO naming conventions apply for *dataset_name*. The target data set must be a sequential data set and must be large enough to accommodate the XQUERY reports you want to collect. You can specify DATASET with or without DISP; the default is DISP(OLD).

  Allocate a data set for the XQUERY function with the following attributes:

```
DCB=(RECFM=FB,LRECL=80,BLKSIZE=6400,DSORG=PS)
```

DISP Specifies how XRC writes the XQUERY output to the data set, as follows:

  MOD Specifies that XRC append the output to the data set.

  OLD Specifies that XRC clear the data set before it receives output.

  SHR Specifies that XRC clear the data set before it receives output. SHR also allows multiple allocations of the same data set.

  If you specify DISP, you must also specify DATASET.

CONFIGURATION
: Specifies that XRC generate a configuration report to provide the details of the XRC session configuration. CONFIGURATION is mutually exclusive with the SET, STORAGECONTROL, ENVIRONMENT, and VOLUME parameters (VOLUME, however, is a valid subparameter of CONFIGURATION). If you specify only the CONFIGURATION parameter, XRC generates a report on the entire configuration.

  You can use combinations of the ERRORLEVEL, SSID, and VOLUME parameters on a single configuration report. The report will include the volumes that meet *all* of the specified filters.

  ERRORLEVEL
  : Specifies that the requested report display only volumes that are defined with the specified error level (session, volume, or *group_name*). If you do not specify ERRORLEVEL, the requested report will contain a list of all volumes which meet the specified report criteria.

  SSID Specifies that the requested report display only volume pairs that are associated with the storage control that matches the specified SSID. If you do not specify SSID, the requested report contains a list of all volumes that meet the specified report criteria. You can request a report that is based on the primary or secondary SSID.

VOLUME

Specifies that the requested report display only volumes that meet the specified volume serial number criteria, as well as any other specified report criteria. The VOLUME parameter has the following options:

- VOLUME(*volser*) provides a report of the primary or secondary volume or list of volumes that meet the specified criteria.
- VOLUME(*volser**) provides a report of the primary volumes that match a single volume serial number prefix. Specify the prefix with one to five characters that are followed by an asterisk (wildcard character).
- VOLUME(ALL) provides a report of all volumes in the session that meet the specified criteria.
- VOLUME(*) provides a report of all volumes in the session that meet the specified criteria and that also match the character and asterisk (wildcard) pattern specified with (*).

You can specify a maximum of 100 primary or secondary volume serial numbers on a single command. If any volumes in the list are not part of the XRC session, XRC includes the volumes in the report with an indication of this status.

STATUS

Specifies that the requested report display only the volumes with a current status that matches the status qualifier. You may select only one option per report. The options are as follows:

OFL Specifies that XRC display offline volumes.

ONL Specifies that XRC display online volumes.

EXTENDER

Specifies a report for storage controls that are connected to a channel extender that provides performance statistics. The standard report shows one line of information for each storage control. EXTENDER is mutually exclusive with SET, VOLUME, CONFIGURATION, SWAP, STORAGECONTROL and ENVIRONMENT.

DETAIL

Specifies that all of the available performance information be displayed.

SET Specifies a report on the current operational values for the XRC session, which are stored in the state data set. SET is mutually exclusive with the STORAGECONTROL, VOLUME, ENVIRONMENT, and CONFIGURATION parameters.

STORAGECONTROL

Specifies a report for all primary storage controls that have volumes in the XRC session. STORAGECONTROL is mutually exclusive with the SET, VOLUME, ENVIRONMENT, and CONFIGURATION parameters.

FEATURES

Specifies that the requested report display the Licensed Internal Code (LIC) features available to the storage controls.

XFEATURES
Specifies that the requested report display the Licensed Internal Code (LIC) features available to the storage controls. This should be used instead of FEATURES for newer feature codes.

DETAIL
Specifies that the requested report display information about standard storage controls or Enhanced (Multiple) Reader primary and auxiliary storage controls individually.

VOLUME
Specifies that the requested report display only volumes that meet the specified volume serial number criteria, as well as any other specified report criteria. For example, specifying:

- VOLUME(*volser*) displays the primary or secondary volume or list of volumes that meet the specified criteria.
- VOLUME(*volser**) displays the primary volumes that match a single volume serial number prefix. Specify the prefix with one to five characters that are followed by an asterisk (wildcard character).
- VOLUME(ALL) displays all volumes in the session that meet the specified criteria.
- VOLUME(*) displays all volumes in the session that meet the specified criteria and that also match the character and asterisk (wildcard) pattern specified with (*).

You can specify a maximum of 100 primary or secondary volume serial numbers on a single command. If any volumes in the list are not part of the XRC session, XRC includes the volumes in the report with an indication of this status. VOLUME is mutually exclusive with the SET, STORAGECONTROL, ENVIRONMENT, and CONFIGURATION parameters.

You can use combinations of the COMMAND, ERRORLEVEL, SCSESSION, SSID, and STATUS parameters on a single volume report. The report will include the volumes that meet *all* of the specified filters.

ACTIVITY
Specifies that a volume report provides the results of a mirror status verification, including the status of all volumes in the session, the consistency timestamp, and the size (in cylinders) of those in duplex. See "Using mirror status verification to check for non-duplex volumes" on page 258 for more information.

DETAIL
Specifies that a volume report provides specific detailed information about device activity level.

PACE Specifies that a volume report provides statistics on device blocking, write pacing, and primary volume write activity for the most recently completed monitor interval. Statistics for write pacing and activity are displayed only for those storage controls with Licensed Internal Code that supports the write pacing function.

Note: When workload-based write pacing is in use, the XQUERY VOLUME PACE report may have several times as many lines it does when workload-based write pacing is not in use.

COMMAND
Specifies that XRC report volumes based on the last pending command issued for the volume, as follows:

ALLCOMMANDS
Specifies that XRC report all volumes that have a pending XDELPAIR or XSUSPEND command request.

NOCOMMANDS
Specifies that XRC report all volumes that do *not* have a pending XDELPAIR or XSUSPEND command request.

ERRORLEVEL
Specifies that the requested report display only volumes that are defined with the specified error level (session, volume, or *group_name*). If you do not specify ERRORLEVEL, the requested report contains a list of all volumes that meet the specified report criteria.

SCSESSION
Specifies that the requested report display volumes that are associated with the storage control session that is specified by *session_name*.

SSID Specifies that the requested report display only volume pairs that are associated with the storage control that matches the specified SSID. If you do not specify SSID, the requested report contains a list of all volumes that meet the specified report criteria. You can request a report that is based on the primary or secondary SSID.

STATUS
Specifies that the volume report display only the listed volumes with a current status that matches the status qualifier. You may select only one option per report. The options are as follows:

BLK Specifies that XRC display device-blocked volumes.

Note: If you specify ACTIVITY in the VOLUME parameter, the STATUS (BLK) is ignored.

BK2 Specifies that XRC display volumes that are not being blocked, with a volume residual count greater than 64 times the value of WrtPacingResidualCnt in PARMLIB member ANTXIN00. Refer to "ANTXIN00 parmlib parameters" on page 126 for more information.

CPY Specifies that XRC display copy volume pairs.

DUP Specifies that XRC display duplex volume pairs.

NON Specifies that XRC display volumes that are not duplex volume pairs. That is, copy, pending, and suspended volume pairs are displayed.

PND Specifies that XRC display pending volume pairs.

SUS Specifies that XRC display suspended volume pairs.

SEQ Specifies that XRC display seqcheck volume pairs.

UTL Specifies that XRC display utility volumes that you have added with the XRCUTL parameter.

ENVIRONMENT
Specifies to display the current values for fields that you can change using the parmlib support. The ENVIRONMENT parameter cannot be used with the VOLUME, SET, STORAGECONTROL, and CONFIGURATION parameters. The values for the ENVIRONMENT parameter are:

FLAG Specifies that the query display all changeable values. There are two ways to change FLAG values. The first method is through the use of parmlib support. The second method is through the MVS MODIFY command.

PARM Specifies that the query display the current settings for the parmlib parameter values. When issued against an inactive or undefined session, the default values for the parameters are displayed.

PATCH
Specifies that the query display the current settings for each patch. When issued against an inactive session, the default values for the patch names are displayed.

MSGROUTEID
Specifies the user ID to which XRC messages that are associated with the processing of this command are routed. If the specified user ID is not logged on, TSO saves messages in the TSO BRODCAST data set and displays them after the next logon.

Note: Messages issued by the initial command parser routine are sent to the user ID that issues the TSO command. If you specify both DATASET and MSGROUTEID, XQUERY output is directed to the data set that is specified with DATASET and not to the MSGROUTEID user ID. Error messages are directed to the MSGROUTEID user ID.

SWAP Specifies that the requested report should display the host volume information (XRC Primary), swap volume information, volume state, and estimated percentage of data on the volume which would be copied to the XRC secondary if a HyperSwap and incremental resynchronization were to be performed.

You can use combinations of the ERRORLEVEL, SSID, and VOLUME parameters on a single swap report. The report will include the volumes that meet *all* of the specified filters.

ERRORLEVEL
Specifies that the requested report display only volumes that are defined with the specified error level (session, volume, or *group_name*). If you do not specify ERRORLEVEL, the requested report will contain a list of all volumes which meet the specified report criteria.

SSID Specifies that the requested report display only volume pairs that are associated with the storage control that matches the specified SSID. If you do not specify SSID, the requested report contains a list of all volumes that meet the specified report criteria. You can request a report that is based on the primary or secondary SSID.

VOLUME
Specifies that the requested report display only volumes that meet the specified volume serial number criteria, as well as any other specified report criteria. The VOLUME parameter has the following options:

- VOLUME(*volser*) provides a report of the primary or secondary volume or list of volumes that meet the specified criteria.
- VOLUME(*volser**) provides a report of the primary volumes that match a single volume serial number prefix. Specify the prefix with one to five characters that are followed by an asterisk (wildcard character).
- VOLUME(ALL) provides a report of all volumes in the session that meet the specified criteria.
- VOLUME(*) provides a report of all volumes in the session that meet the specified criteria and that also match the character and asterisk (wildcard) pattern specified with (*).

You can specify a maximum of 100 primary or secondary volume serial numbers on a single command. If any volumes in the list are not part of the XRC session, XRC includes the volumes in the report with an indication of this status.

STATUS
Specifies that the requested report display only the volumes with a current status that matches the status qualifier. You may select only one option per report. The options are as follows:

OFL Specifies that XRC display offline volumes.

ONL Specifies that XRC display online volumes.

For additional information about:
- XQUERY configuration, XQUERY SET, and XQUERY storage control reports and to view an example of each report, refer to“Using XRC reports” on page 184.
- Parmlib support, refer to Chapter 6, “Administering your extended remote copy environment,” on page 123.
- How to use the MVS MODIFY command, refer to “Diagnosing system data mover functions with the MVS MODIFY command” on page 549.
- MVS support, refer to Appendix A, “Advanced Copy Services diagnostic aids,” on page 549.

XQUERY command examples

The following are examples of the XQUERY command. If you do not use single quotation marks, TSO by default prefixes the user ID to the data set name.

```
XQUERY MAINZ VOLUME(ALL) DETAIL STATUS(DUP)

XQUERY MAINZ VOLUME(flr*) DATASET('XQUERY.OUTPUT1') DISP(MOD)

XQUERY TUCRUN CONFIGURATION ERRORLEVEL(SESSION) STATUS(CPY)
       COMMAND(NOCOMMANDS)
```

XQUERY MASTER in a coupled environment

Issue the XQUERY MASTER command to a master session to display information about coupled or clustered XRC sessions associated with the master session, including the XRC session name, session status, command pending status, journal delta time, RCV/ADV delta time, master session status, master session recoverable time, and master pending command.

XQUERY MASTER in a coupled environment command syntax: The syntax of the XQUERY MASTER command in a coupled environment is :

```
►►──XQUERY──msession_id──MASTER──┬─────────────────────────┬──────────────────────────────►
                                 │           ┌─SYS1─┐      │
                                 └─MHLQ──(───┴─mhlq─┴───)──┘

►──┬──────────────────────────────────────────────────┬──┬──────────────────────────────┬──►◄
   └─DATASET──(──dataset_name──)──┬──────────────────┬┘  └─MSGROUTEID──(──user_ID──)──┘
                                  │         ┌─OLD─┐  │
                                  └─DISP──(─┼─MOD─┼─)┘
                                            └─SHR─┘
```

Required parameters:

msession_id
Specifies the name of the master session. The master session is a logical entity that is used to coordinate session commands and data consistency across multiple XRC sessions. A master session exists as long as there is an XRC session coupled to the master session. The maximum length of the name is eight characters. Do not include embedded blanks. TSO ignores leading blanks and trailing blanks.

Optional parameters:

DATASET
Specifies that XRC direct the XQUERY output to the sequential data set that is identified by *dataset_name*. Standard TSO naming conventions apply for *dataset_name*. The target data set must be a sequential data set and must be large enough to accommodate the XQUERY reports you want to collect. You can specify DATASET with or without DISP; the default is DISP(OLD).

DISP Specifies how XRC writes the XQUERY output to the data set, as follows.

MOD Specifies that XRC append the output to the data set.

OLD Specifies that XRC clear the data set before it receives output.

SHR Specifies that XRC clear the data set before it receives output. SHR also allows multiple allocations of the same data set.

If you specify DISP, you must also specify DATASET.

MASTER
Specifies that XQUERY produce a new master coupling report. It may be specified in conjunction with the DATASET and MSGROUTEID parameters.

MHLQ*(mhlq)*
Specifies the high-level qualifier of the master session control data set. The default is SYS1.

MSGROUTEID
Specifies the user ID to which XRC messages that are associated with the processing of this command are routed. If the specified user ID is not logged on, TSO saves messages in the TSO BRODCAST data set and displays them after the next logon.

Note: Messages issued by the initial command parser routine are sent to the user ID that issues the TSO command. If you specify both DATASET and MSGROUTEID, XQUERY output is directed to the data set that is specified with DATASET and not to the MSGROUTEID user ID. Error messages are directed to the MSGROUTEID user ID.

XQUERY MASTER in a coupled environment command examples: The following are examples of the XQUERY MASTER in a coupled environment command:

```
XQUERY HOSPITAL MASTER MHLQ(SYS2) DATASET (XQUERY.OUTPUT2) DISP(OLD)

XQUERY SCHOOLS MASTER MHLQ(SYS1) MSGROUTEID(QADAMS)
```

XRECOVER–Recovering data on the recovery system

Use the XRECOVER command to bring data on the XRC recovery system to a consistent, recoverable state in the event of a disaster to the primary system.

The first command at the recovery site must be the XRECOVER command. You will receive a "session not active" message if you issue another XRC command before an XSTART or XRECOVER command has completed.

When you issue the XRECOVER command to each individual XRC session, it determines the common consistency time to which all coupled sessions can be recovered. All appropriate updates are then applied to the target XRC session to reach that consistency time.

Issue the XRECOVER command on the recovery system to complete the updating of all journal data to the secondary (target) volumes. The secondary volume serial numbers of volume pairs in the duplex or seqcheck state are changed to match ("clipped to") the primary volume serial number. The secondary volume serial numbers of suspended volume pairs are also clipped, provided that the pairs have completed initial volume synchronization prior to being suspended. For SESSIONTYPE(XRC), the XRC recovery function must continue to have access to the appropriate journal, control, and state data sets that were in use at the time of the failure, or when the XRC session ended. A migration session requires only the state data set.

If the XRC session has either been suspended or ended normally by a command, and the FORCE keyword has not been specified on the XRECOVER command, processing compares the internal timestamps of the state and control data sets of the XRC session being recovered with the internal timestamps of the master and cluster data sets to ensure that the difference between them is less than the value for DEADSESSIONDELAY in the ANTXIN00 member of PARMLIB plus 15 seconds. Additionally, if the XRC session is coupled, processing checks that the number of volumes being included in the XRECOVER matches the number of volumes expected to be recovered in the coupled or clustered environment. The purpose of these checks is to ensure that the correct control data sets are being used for the XRECOVER.

Next, XRECOVER applies the journaled data to update secondary volumes that were suspended while XRC was writing the last consistency group when the XRC session was active. XRC attempts to apply all outstanding data to the secondary volumes so that these volumes are consistent with the rest of the session.

If the suspended volumes can be made consistent, XRC then changes the volume status from SUS to RCV and clips the secondary. If the XRC session has continued to apply consistency groups, then the volume stays in SUS status. XRC relabels the secondary volumes to the primary volume serial numbers, provided that the volume pair had completed initial volume synchronization prior to the recovery action.

You can also issue the XRECOVER command to recover a group of interlocked coupled sessions to the same consistency time.

In order to provide master session consistency, CXRC applies updates to volumes on the sessions that are behind to allow them to advance to the forward session. CXRC will forward-recover volumes that are suspended because of an error and that have the necessary journal data available. It will *not* forward-recover volumes that are suspended because of an XSUSPEND command.

Finally, the XRECOVER command automatically generates a recovery report. The report includes the last applied timestamp that defines the recovered, consistent state for all volumes. The XRECOVER command also automatically generates an XQUERY MASTER report for coupled sessions.

Note: Do not issue the XRECOVER command to an active XRC session. If a session is active when you issue the XRECOVER command, XRC rejects the command. You must first issue an XSUSPEND or XEND command for the session, and then make all primary volumes offline to the SDM system. The XRECOVER command that follows the XSUSPEND or XEND command relabels each eligible secondary volume with the primary volume serial number.

For more information:

- About how to use the XRECOVER command, refer to Chapter 9, "Extended remote copy data recovery operations," on page 241
- About how to use the XRECOVER command in a coupled environment, refer to "Example: recovering from a disaster using the XRECOVER command" on page 237
- About the DEADSESSIONDELAY parameter in the ANTXIN00 member of PARMLIB, refer to "ANTXIN00 parmlib parameters" on page 126.

XRECOVER command syntax

```
>>──┬─XRECOVER─┬──session_id──┬────────────────────────────────┬──────────────>
    └─XRCV─────┘              │               ┌─SYS1──────┐    │
                              └─HLQ──(──┴─qualifier─┴──)─┘

>──┬─────────────────────────┬──┬──────────┬──┬─────────────────────────┬──┬───────┬──><
   │           ┌─SYS1─┐      │  ├─ONLINE───┤  └─MSGROUTEID──(user ID)─┘  ├─CHECK─┤
   └─MHLQ──(───┴─mhlq─┴──)───┘  └─TERTIARY─┘                             └─FORCE─┘
```

Required parameters:

session_id
: Specifies the name of an XRC session. This can be any name acceptable to TSO that matches the name of a session that was previously specified on the XSTART command. The maximum length of the name is eight characters. Do not include embedded blanks. TSO ignores leading blanks and trailing blanks.

Optional parameters:

CHECK
: Specifies that **only** XRECOVER enhanced checking should be performed, bypassing other XRECOVER processing. This produces an XRECOVER CHECK report. CHECK is mutually exclusive with FORCE.

FORCE
Specifies that the XRECOVER enhanced checking should not be performed. FORCE is mutually exclusive with CHECK.

HLQ Specifies the name of the high-level-qualifier that the XRECOVER command will refer to when you issue it. The XRECOVER HLQ name must be the same name that was specified on the XSTART command that initiated the XRC session, if an HLQ was specified. If you started the session without specifying an HLQ, the XRECOVER HLQ name must be from one to eight characters, and can be any name acceptable to TSO. SYS1 is the default value.

MHLQ*(mhlq)*
Specifies the name of the high-level qualifier for the master session control data set, which was previously specified on the XCOUPLE command. The MHLQ name must be from one to eight characters, and can be any name acceptable to TSO. SYS1 is the default value.

MSGROUTEID
Specifies the user ID to which XRC messages associated with the processing of this command are routed. If the specified user ID is not logged on, TSO saves messages in the TSO BRODCAST data set and displays them after the next logon.

Note: Messages issued by the initial command parser routine are sent to the user ID that issues the TSO command.

ONLINE
Specifies any online volume that matches the volser of a secondary volume can be used for recovering data. See "Accessing secondary volumes while XRC is active" on page 192 for more information.

TERTIARY
Specifies that any online volume that matches the volume serial of a secondary volume can be used for recovering data, as long as that volume is not the same secondary volume that was in use when XRC was suspended or ended.

XRECOVER command example

The following is an example of the XRECOVER command:

```
XRECOVER MAIN1 MSGROUTEID(OPER9)

XRECOVER session FORCE HLQ(SYS1) MHLQ(MSESSHLQ)

XRECOVER session CHECK
```

XSET–Changing session parameters

Use the XSET command to change the characteristics of an active session without requiring that you suspend and restart the session. The XSET command allows you to dynamically change XRC session control values.

When the XSET command processes, characteristics of the XRC session are updated. The changes remain in effect for the duration of the session, even through session suspension and restart, unless changed with another XSET command.

XSET command syntax

```
►►──XSET──session_id──┬──────────────────────────┬──┬──────────────────────────┬──────────────►
                      │          ┌─FULL──┐       │  │             ┌─ 8 ──┐     │
                      └─COPY──(──┴─QUICK─┴──)────┘  └─PAGEFIX──(──┴─ffff─┴──)──┘

►──┬───────────────────────────────────────────────────────────────────────────────────┬──────►
   └─PARMLIB──(──member_name──ACTION──(──action──)──DATASET──(──parmlib_dataset_name──)──)─┘

►──┬───────────────────────────┬──┬─────────────────────────────────┬──┬──────────────────────────┬──►
   │             ┌─LOAD─┐      │  │                ┌─00.30.00─┐     │  │            ┌─7500──┐     │
   └─PRIORITY──(─┴─FIFO─┴──)───┘  └─RFREQUENCY──(──┴─hh.mm.ss─┴──)──┘  └─RTRACKS──(─┴─ttttt─┴──)──┘

►──┬──────────────────────────────┬──┬──────────────────────────────┬──┬─────────────────────┬──►
   │            ┌─2,2─┐           │  │              ┌────────────┐  │  │          ┌─ 4 ┐     │
   └─SCSYNCH──(─┬┴─p,s─┴─┬──)─────┘  │              ▼  ┌─NO──┐   │  │  └─SYNCH──(─┴─ss─┴──)─┘
                │ ┌─2─┐  │           └─SUSLBUSY──(───┴─YES─┴───┴─)─┘
                └─┴─n─┴──┘

►──┬────────────────────────────────────────────────────────────────────────┬──────────────►
   │ ┬─REFRESHP──────────────────────────────────────┬ ┬────────────────────────────┬
   │ └─TIMEOUT──(──┬─hh.mm.ss────────────────────┬──)┘ │            ┌─────────┐     │
   │               └─STORAGE_CONTROL_DEFAULT─────┘     │            ▼         │     │
   │                                                   └─SSID──(──┬─ssid─┬───┴──)──┘
   │                                                              └─ALL──┘

►──┬────────────────────────────┬──┬─DVCBLOCK──(──┬─ON──┬──)──VOLUME──(──┬─volume_list─┬──)─┬──►
   │            ┌─FLOAT─┐       │                 ├─OFF─┤                ├─ALL─────────┤
   └─UTILITY──(─┴─FIX───┴──)────┘                 └─WPn─┘                ├─ALLDP───────┤
                                                                         ├─ALLP────────┤
                                                                         └─ALLB────────┘

►──┬──────────┬──┬──────────────────────────┬─────────────────────────────────────────────────►◄
   └─REFRESHS─┘  └─MSGROUTEID──(user ID)────┘
```

Required parameters:

session_id
: Specifies the name of an XRC session. This can be any name acceptable to TSO that matches the name of a session that was previously specified on the XSTART command. The maximum length of the name is eight characters. Do not include embedded blanks. TSO ignores leading blanks and trailing blanks.

  Note: When the PARMLIB parameter is specified on the XSET command, the name of a cluster can be specified as the session name . For all other parameters, you cannot specify a cluster name as the session name.

Optional parameters:

COPY
: Specifies the default initialization method for the secondary volumes of volume pairs that are added to an XRC session.

  The values are:

  FULL
  : Specifies that XRC make a copy of each track of the primary volume onto the secondary volume at the same time that additional application updates are made to both copies. The volumes are initially in pending status and enter duplex state when the initial volume copy is completed.

  QUICK
  : Specifies that XRC make a copy of only allocated tracks on the primary volume onto the secondary volume at the same time that additional application updates are made to both copies. XRC reads the VTOC of the

primary volume to determine the allocated tracks on the volume. The volumes are initially in pending status and enter duplex state when the quick volume copy completes.

PAGEFIX

Specifies the maximum amount of real storage that XRC will keep page fixed to process I/O operations. This storage is freed back to the system when the system data mover no longer needs the storage. The system data mover, however, can fix and free any amount of additional real storage, for short-term durations, as required, up to 35 MB per storage control session.

The default value is 8 MB. You can specify the value from zero (no permanent page-fixed storage) to 9999. Changes specified with the PAGEFIX parameter take place when the next set of storage control buffers are processed.

Attention: Unpredictable results can occur if you specify an amount of permanently page-fixed storage that exceeds or is near the limits of available processor storage. An IPL may be necessary.

PARMLIB

To invoke parmlib support at times other than XRC start up, you can issue the XSET PARMLIB command. You can use the XSET PARMLIB command both before and after you issue an XSTART command. If you invoke XSET PARMLIB before an XSTART command, you can check the validity of the parameter syntax without applying any of the parameters. Syntax checking looks for occurrences of data specification errors, such as missing commas and illegal parameters, as well as the validity of the commands and their structure. If you invoke XSET PARMLIB after an XSTART command, you can apply the command parameters as well as perform syntax checking.

The following values must be specified as noted when using the XSET PARMLIB command:

member_name

Specifies which parmlib member to read.

ACTION*(action)*

Specifies the type of action to perform. This can be:

- VERIFY, which performs a syntax check. This is the default.
- APPLY, which performs a syntax check, and if everything is correct, applies the parameters.

For information about limitations to using APPLY with changes to WorkloadWritePacing, refer to "ANTXIN00 parmlib parameters" on page 126.

DATASET*(parmlib_dataset_name)*

Specifies the parameter library to use. This field is optional. If you do not supply a value for this field and the specified session is active, XRC uses the data set that is set up when the XSTART command processes.

XRC requires this parameter if the session (*session_id*) is not an existing session.

Note: If *session_id* is not an active logical session, XSET runs in address space ANTAS000. XRC only performs a verification (return code 4088 if ACTION is set to apply), and the DATASET parameter is required (return code 4089 if DATASET is missing). This process provides a way of checking parameters before an XSTART command is issued.

PRIORITY
Specifies the priority that the XADDPAIR command uses for selecting the next volume to synchronize or resynchronize. Changes specified with the PRIORITY parameter take place when the next volume is processed. The default value is LOAD.

The values are:

FIFO
Specifies that the system data mover select volumes in the order that is specified on the XADDPAIR command.

LOAD
Specifies that the system data mover select volumes whose primary storage control has the least load.

RFREQUENCY
Specifies how long the system data mover will wait before it resets the resynchronization bitmaps. You can specify the frequency in hours, minutes, and seconds. The time can range from 00.00.30 to 18.00.00. The default time is 30 minutes. If you set the frequency value to zero (00.00.00), XRC does not use elapsed time to determine whether to reset the storage control session bitmap.

RTRACKS
Specifies the number of tracks that must change before the system data mover resets the resynchronization bitmaps. You can specify a value from 0 to 99999. The default value is 7500 tracks.

SCSYNCH
Specifies the maximum number of volume pairs that the system data mover can synchronize or resynchronize concurrently per storage control.

You can specify the XSET SCSYNCH parameter as SCSYNCH(*p,s*) or as SCSYNCH(*n*). The limit specified with *n* applies to both primary and secondary storage controls. You can specify a value between 0 and 45 for each variable (*p,s,* or *n*).

Examples: These are possible examples for setting the XSET SCSYNCH parameter:
- `XSET SCSYNCH(2)`
- `XSET SCSYNCH(2,2)`
- `XSET SCSYNCH(4,2)`

Set the SCSYNCH value in conjunction with the SYNCH parameter, which specifies the *session* limits for concurrent volume synchronization tasks. Changes specified with the SCSYNCH parameter take place when the next volume initialization is processed.

SUSLBUSY
Used to enable or disable the Suspend on Long Busy function. Acceptable values are YES (enable) and NO (disable), with NO as the default.

SUSLBUSY is mutually exclusive with TIMEOUT and REFRESHP. If SSID is also specified, the attributes of the storagecontrol sessions associated with the specified SSID's are immediately modified. If SSID is not specified, the global value for the session is modified. The global value is used for any new storage control session that is subsequently added.

If the storage control microcode supports the function, enabling will cause the microcode to automatically suspend the storage control session instead of raising extended long busy when sidefile limits are exceeded.

If the storage control microcode does not support the function, enabling will cause the data mover to suspend a storage control session as soon as it detects extended long busy due to sidefile exceeding limits. Storage control sessions that do not support suspension are terminated

Disabling allows the data mover to tolerate the long busy condition for 80% of the storage control session timeout interval, after which mirroring is suspended.

SYNCH
Specifies the maximum number of volume synchronization or resynchronization tasks that XRC can concurrently start in the XRC session. The range of *ss* is from zero to 45 tasks; the default is four. Specify a value of zero to ensure that XRC does not start any new volume initialization. XRC will continue with existing volume synchronization or resynchronization tasks. Changes specified with the SYNCH parameter take place when XRC processes the next volume. SYNCH specifies an XRC session-level value. For each synchronization task that it starts, XRC fixes real page storage that is based on the following formula:

```
{ SYNCH * (number of volume pairs in CPY status) * 360K }
```

TIMEOUT
Specifies the primary storage control timeout value for application impact. This value specifies the maximum time that applications are unable to update volumes before the storage control suspends, or ends, the storage control session.

You can specify this parameter in hours, minutes, and seconds, can range from a minimum of one second (00.00.01) to a maximum of 18 hours (18.00.00). Specify TIMEOUT(STORAGE_CONTROL_DEFAULT) to request that XRC use the default set in the storage control, which is normally five minutes (00.05.00). Changes specified with the TIMEOUT parameter take place when you add a new storage control to the XRC session. You can also specify which storage subsystems are immediately affected by the TIMEOUT parameter, using the SSID parameter.

TIMEOUT is mutually exclusive with SUSLBUSY.

SSID
Specifies the SSIDs that the value specified on the TIMEOUT is applied to, thereby qualifying the scope of the TIMEOUT parameter.

If you do not specify the SSID parameter, then only storage controls that are added in the future will get the new timeout value. If you specify SSID(ALL), all current and future storage controls immediately get the new timeout value.

If you specify a specific set of SSIDs, only the listed storage controls immediately use the new timeout value. Storage controls that you add in the future will get the original timeout value, not the one specified in the TIMEOUT parameter.

You can specify up to nine SSIDs on a single XSET command. Specify each SSID with up to four character values. Separate each SSID by a comma or a blank.

UTILITY
Specifies how the utility device is selected for reading data from the primary storage control. The values include:

FLOAT
Specifies the utility device for a storage control session will be selected by the storage control. FLOAT is the default.

FIX
Specifies the utility device for a storage control session is to be a specified (fixed) device.

Note: Be aware that issuing the XSET command with the UTILITY parameter specified does not activate or deactivate fixed utility device support. This support is enabled when you issue an XADDPAIR command with a secondary volser of XRCUTL.

DVCBLOCK
Specifies the device blocking option to be applied to the volume, list of volumes, or all volumes in a session specified with the VOLUME parameter.

Workload-based write pacing affects the behavior of DVCBLOCK. For more information, refer to "Workload-based write pacing" on page 161.

The DVCBLOCK values are:

ON Specifies that device blocking is enabled for the specified volume(s).

When workload-based write pacing is in use, XRC automatically converts DVCBLOCK(ON) to DVCBLOCK(WP*n*), where *n* is the discretionary level, 6.

OFF
Specifies that device blocking is disabled for the specified volume(s). This is the default.

WP*n*
Specifies that write pacing is to be activated for the specified volume(s), with *n* specifying the level of write pacing delay, 0-F.

WP0 specifies that the session default level will be used, as specified the SHADOW DfltWritePacingLvl PARMLIB value.

WP1-WP7 result in pacing maximums of 0.02, 0.04, 0.1, 0.5, 1, and 2 milliseconds per recordset, respectively. These levels are useful for volumes with high rates of small blocksize writes, such as data base logs, where minimal response time impact is essential.

WP8-WPC result in pacing maximums of 5, 10, 25, 50, and 100 milliseconds per recordset, respectively. These levels are useful for volumes with high mb/sec write rates.

WPD-WPF result in pacing maximums of 200, 500, and 1000 milliseconds per recordset, respectively. These levels should be used only in exceptional situations where a very high degree of pacing is required.

Delay is injected per recordset, but a write channel program might create several recordsets. In such cases, the maximum possible delay per write channel program is equal to the pacing maximum multiplied by the number of recordsets that the channel program creates. For example, at pacing level WPC, a channel program that creates 4 recordsets can be delayed for a total of 4×100, or 400 milliseconds.

There is no overall limit on the amount of delay that can be injected for a channel program. Very large channel programs, such as those used in Sort applications, can experience delays measured in seconds when higher pacing levels are used.

Note: Only the volumes specified with the VOLUME parameter are affected by the DVCBLOCK parameter. No other XSET parameters are associated with this parameter.

VOLUME
Specifies that all volumes, a list of volumes, or a single volume in a session be processed according to the value specified in the DVCBLOCK parameter.

The VOLUME values are:

volume_list
A list of 1 to 50 six-character volume serial numbers. Use either a blank of a comma to separate the serial numbers.

ALL
Specifies that all volumes in a session are to be changed. Message ANTX8131I will be issued indicating the total number of volumes for which the DVCBLOCK value was accepted.

ALLDP
All primary volumes specified as WP0 are to be changed.

ALLB
All primary volumes set to DVCBLOCK(ON) are to be changed.

ALLP
All primary volumes specified as WP1 - WPF are to be changed.

REFRESHP
Specifies that the system data mover is to examine the capabilities of the specified primary storage control(s) and update the data mover's internal control information accordingly. This allows detection of new capabilities, such as write pacing, without having to suspend or reissue the addpair request.

The system data mover doers not compare the data on the old and new volumes.

REFRESHP cannot be used with TIMEOUT.

REFRESHS
Specifies that the system data mover is to examine the SSID and CCA locations of any suspended secondary volume and update the system data mover's internal control information. If you make an exact copy of one or more secondary volumes, you can resume mirroring with the new volumes without first having to reissue XDELPAIR for any of the old volumes and XADDPAIR for any of the new volumes. This option thus eliminates the need for a full copy from primary to secondary volume, resulting in a faster return to duplex mode.

The system data mover does not compare the data on the old and new volumes. You must ensure that the relocated secondary volume is identical to the original secondary at the time of suspension.

The following conditions must exist to update the location of any secondary volume:

- The session must be active.
- Each pair with secondary volumes to be relocated must be suspended.
- Each new secondary volume must be online to the system data mover system.
- Each new secondary volume must have the same volser, the same or greater number of cylinders, and the same number of tracks per cylinder as the old

secondary volume. If a secondary volume is relocated to a larger secondary volume, you cannot switch back to the old secondary.

- Ensure that the XRC PARMLIB parameter VOLINIT ENABLEREFRESHS(YES) is specified.

Using a combination of XRC commands, you can track which secondary volumes are relocated and where they were relocated:

1. Issue an XQUERY CONFIG request, redirecting it to a data set.
2. Issue XSET REFRESHS. You receive message ANTX8141I indicating number of volumes relocated.
3. Issue the XQUERY CONFIG request to redirect the data set to a different data set.
4. Compare the two XQUERY CONFIG data sets to find the relocated secondary volumes.

Note:

1. Use REFRESHS with care to prevent creating inconsistent data on the primary and secondary volumes.
2. REFRESHS cannot be used with any other optional XSET keyword except MSGROUTEID.

MSGROUTEID

Specifies the user ID to which XRC messages associated with the processing of this command are routed. If the specified user ID is logged off, TSO saves messages in the TSO BRODCAST data set and displays them after the next logon.

Note: Messages issued by the initial command syntax-checking routine are sent to the user ID that issues the TSO command.

For additional information about:

- Using the XSET COPY command and the InitializationMethod parameter, refer to “ANTXIN00 parmlib parameters” on page 126.
- Using the XSET PAGEFIX command the PermanentFixedPages parameter, refer to “ANTXIN00 parmlib parameters” on page 126.
- Using the XSET PARMLIB command, refer to “XSET PARMLIB” on page 126.
- Using the XSET PRIORITY command and the SelectionAlgorithm parameter, refer to “ANTXIN00 parmlib parameters” on page 126.
- Using the XSET RFREQUENCY command and the DelayTime parameter refer to “ANTXIN00 parmlib parameters” on page 126.
- Using the XSET RTRACKS command and the ChangedTracks parameter, refer to “ANTXIN00 parmlib parameters” on page 126.
- Using the XSET SCSYNCH command and the InitializationsPerPrimary and InitializationsPerSecondary parameters, refer to “ANTXIN00 parmlib parameters” on page 126.
- Using the XSET SYNCH command and the MaxNumberInitializations parameter, refer to “ANTXIN00 parmlib parameters” on page 126.
- Using the XSET TIMEOUT command and the StorageControlTimeout parameter, refer to “ANTXIN00 parmlib parameters” on page 126 and “Using the XSET TIMEOUT parameter” on page 183.
- Using the XSET UTILITY command refer to “ANTXIN00 parmlib parameters” on page 126.

- XRC utility devices, refer to "Using XRC utility devices" on page 165.

XSET command examples

The following are examples of the XSET command:

```
XSET  OLYMPIC  PAGEFIX(70)  PRIORITY(FIFO)  SCSYNCH(3)
      SYNCH(8)  TIMEOUT(00.40.00) MSGROUTEID(OPER1)

XSET  PANDA  PAGEFIX(140)  PRIORITY(LOAD)  SYNCH(8) TIMEOUT(01.30.00)
  RTRACKS(18000) FREQUENCY(04.00.00)  MSGROUTEID(KING9) SSID(ALL)

XSET PANDA UTILITY(FIX) PAGEFIX(12)
```

XSTART–Starting a session

Use the XSTART command to:

- Start a new XRC session
- Restart a suspended XRC session

When the suspended XRC session becomes active, the suspended volumes are not automatically resynchronized. You have to issue an XADDPAIR command to resynchronize all volume pairs within the session. You can set up the XSTART command to automatically start the session when you initialize the system.

See "Example: coupling new or existing XRC sessions" on page 220 for a scenario that shows you how to use XSTART to couple an XRC session to a master session.

XSTART command syntax

```
>>—XSTART—session_id——ERRORLEVEL——(——SESSION————)——————————————————>
                      └ERRLVL————┘   ├VOLUME—————┤
                                     └group_name—┘

>—SESSIONTYPE—(——MIGRATE——)————————————————————————————————————————>
                └XRC——————┘  └HLQ—(——┬SYS1┬——)┘
                                     └hlq─┘

>——————————————————————————————————————————————————————————————————><
  └MSGROUTEID—(user_ID)┘
```

Required parameters:

session_id
: Specifies the name of an XRC session. This can be any name acceptable to TSO except for the word ALL, which is reserved. The maximum length of the name is eight characters. Do not include embedded blanks. TSO ignores leading blanks and trailing blanks.

ERRORLEVEL
: Specifies how the XRC session responds when an error occurs that causes the session to become unable to process a volume. You can abbreviate this parameter as **ERRLVL**.

  The error level you specify with the XSTART command is the default for all volumes that do not have an error level specified with an XADDPAIR command. The error level specified with the XADDPAIR command overrides the error level specified with the XSTART command. For example, if you specify ERRORLEVEL(SESSION) for a volume pair on an

XADDPAIR command, an error on that duplex volume pair results in the suspension of all volumes in the XRC session, even if ERRORLEVEL(VOLUME) was specified on the XSTART command.

XRC supports multiple levels of error processing within a single session. Critical volumes can have ERRORLEVEL(SESSION), while all other volumes can use ERRORLEVEL(VOLUME) or ERRORLEVEL(*group_name*).

The values are:

SESSION

Specifies that, if a permanent error to a duplex primary or secondary volume occurs, XRC suspends *all volume pairs* in the XRC session, regardless of the volume status. Suspending any volume pairs from the session can negatively affect the usability of the remaining secondary volumes. Specify SESSION to ensure that all secondary volumes in the entire XRC session that are necessary for recovery are consistent up to the time of failure.

For sessions that will be coupled, use ERRORLEVEL(SESSION) to ensure a recoverable consistency across all coupled sessions.

VOLUME

Specifies that, if a permanent error occurs on a duplex volume, the XRC session suspends the volume pair or pairs that are associated with the error. All other volumes continue to process.

group_name

Specifies that, if an error to a duplex primary or secondary volume pair occurs, XRC suspends all volume pairs that are associated with that group name.

The *group_name* can be any name acceptable to TSO. The maximum length of the name is eight characters. Do not include embedded blanks. TSO ignores leading blanks and trailing blanks. Do not use the reserved names VOLUME and SESSION.

Note: For coupled sessions, error processing for volumes that are contained in a group will only pertain to the volumes within that session.

SESSIONTYPE

Specifies the operating mode (MIGRATE or XRC) for the XRC session. To change the mode of operation, suspend the session with an XSUSPEND command, followed by an XSTART command with a new SESSIONTYPE parameter. The values are:

MIGRATE

Specifies that the system data mover apply updates to the secondary volumes on a consistency group basis, but not write updates to the journal data sets. You must already have allocated the XRC state data set. The journal and control data sets are not used.

XRC commands maintain data consistency within a migration session in the same way as in a recovery session. If a disaster occurs while in SESSIONTYPE(MIGRATE) mode, XRC cannot guarantee data consistency across the secondary volumes, as there is no journaling activity. You can issue an XRECOVER command after ending a session in MIGRATE mode to clip the secondary volume serial numbers.

XRC Specifies that XRC operate in recovery mode. A SESSIONTYPE(XRC) session makes use of journal volumes to guarantee secondary volume consistency group time. If XRC removes a volume from the session due to an I/O error, XRC can no longer update the secondary volume. Subsequent updates to the primary volume are lost if you must recover the applications on the recovery system.

Optional parameters:

HLQ Specifies the name of the high-level-qualifier for the state, control, and journal data sets to which the XSTART command refers. The XSTART HLQ name must be from one to eight characters, and can be any name acceptable to TSO. SYS1 is the default value.

MSGROUTEID
Specifies the user ID to which XRC messages associated with the processing of this command are routed. If the specified user ID is not logged on, TSO saves messages in the TSO BRODCAST data set and displays them after the next logon.

Note: Messages issued by the initial command parser routine are sent to the user ID that issues the TSO command.

XSTART command example

The following is an example of using the XSTART command:

```
XSTART MAIN1 SESSIONTYPE(XRC) ERRORLEVEL(VOLUME) MSGROUTEID(OPER1)
```

This command:
- Starts an XRC session called MAIN1
- Suspends an XRC volume pair if an error occurs when specifying the ERRORLEVEL(VOLUME)
- Routes XRC messages that result from processing this command to user ID OPER1

XSTATUS – querying XRC status

Use the XSTATUS command to query the status of different extended remote copy (XRC) activities.

XSTATUS command syntax

```
►►──XSTATUS──┬───────────────────────────────────────────────────┬────────►◄
             ├─CLUSTER | ADDRSPACE───────────────────────────────┤
             ├─DATASET──(──dataset_name──)──┬─────────────────┬──┤
             │                              │      ┌─OLD─┐    │  │
             │                              └─DISP─(─┼─MOD─┼─)─┘  │
             │                                     └─SHR─┘       │
             └─MSGROUTEID──(──user_id──)─────────────────────────┘
```

Optional parameters::

CLUSTER | ADDRSPACE
CLUSTER and ADDRSPACE are mutually exclusive required parameters.

CLUSTER

Specifies that XRC display information about coupled sessions associated with the cluster session in the logical partition where the command is processed, including the following information:

- Cluster session name
- XRC session name
- Session status
- Volume status
- Interlock status
- Journal delta time
- RCV/ADV delta time
- Master session name and HLQ
- Maximum journal delta for all sessions in the cluster session
- Maximum RCV/ADV delta time for all sessions in the cluster session

ADDRSPACE

Specifies that XRC display the names of all active XRC address spaces in the logical partition, including the following information:

- Session name
- Address space name and identifier
- Type of session
- Associated cluster name
- Associated master session name

DATASET
Specifies that XRC direct the XSTATUS output to the sequential data set that is identified by *dataset_name.* Standard TSO naming conventions apply for *dataset_name.* The target data set must be a sequential data set and must be large enough to accommodate the XSTATUS reports you want to collect. You can specify DATASET with or without DISP; that default is DISP(OLD).

Allocate a data set for the XSTATUS function with the following attributes:

```
DCB=(RECFM=FB,LRECL=80,BLKSIZE=6400,DSORG=PS)
```

DISP Specifies how XRC writes the XSTATUS output to the data set, as follows:

MOD Specifies that XRC append the output to the data set.

OLD Specifies that XRC clear the data set before it receives output.

SHR Specifies that XRC clear the data set before it receives output. SHR also allows multiple allocations of the same data set.

If you specify DISP, you must also specify DATASET.

HLQ Specifies the name of the high-level-qualifier for the state, control, and journal data sets to which the XSTART command refers. The XSTART HLQ name must be from one to eight characters, and can be any name acceptable to TSO. SYS1 is the default value.

MSGROUTEID
Specifies the user ID to which XRC messages that are associated with the processing of this command are routed. If the specified user ID is not logged on, TSO saves messages to be in the TSO BRODCAST data set and displays them after the next logon.

Note: Messages issued by the initial command parser routine are sent to the user ID that issues the TSO command. If you specify both DATASET and MSGROUTEID, XQUERY output is directed to the data set that is specified with DATASET and not to the MSGROUTEID user ID. Error messages are directed to the MSGROUTEID user ID.

XSUSPEND–Suspending volumes or sessions

Use the XSUSPEND TSO command to either suspend an XRC session or to suspend one or more volume pairs from the XRC session. In a coupled environment, you may either suspend all XRC sessions coupled to the master session or suspend all volume pairs from sessions that are coupled to the master session.

When an XRC session is suspended, the system data mover is no longer active and the volume pairs are in suspended state. Session suspension should occur as part of a planned outage when you stop applications or intend the outage to be brief. Suspending an XRC session cancels all pending volume commands; you must reissue pending commands when you restart the session.

The system data mover remains active when a volume pair is in suspended state. If a storage control supports a hardware bitmap, the SDM does not read record updates from it. If a storage control does not support a hardware bitmap, the SDM continues to read record updates from the primary storage control that owns the suspended volume. A volume-level suspension does not affect application activity.

After you issue the XSUSPEND command, XRC prompts you (unless you have specified NOVERIFY) to confirm the request to suspend the session or set of specified volumes. If the XSUSPEND command is invoked through a CLIST written in REXX, you should either specify NOVERIFY or place your response to the prompt in the data stack prior to issuing the command. In a batch environment, you can stack the response.

For additional information about suspending coupled sessions, refer to “XSUSPEND in a coupled environment” on page 118.

XSUSPEND command syntax

The syntax of the XSUSPEND command is:

```
►►──XSUSPEND──session_id──┬─TIMEOUT──(──┬─hh.mm.ss────────────────┬──)─┬──────────►
                          │             └─STORAGE_CONTROL_DEFAULT─┘    │
                          │              ┌──────────────┐              │
                          │              ▼              │              │
                          └─VOLUME──(──┬───primary_volser─┴──)─────────┘
                                       └─ALL──────────────┘

   ┌─IMMEDIATE─────────────────────────────────┐
►──┼───────────────────────────────────────────┼──┬──────────┬──────────────────►
   ├─ATTIME──(──yyyy.ddd hh:mm:ss.thmiju──)────┤  └─NOVERIFY─┘
   ├─DRAIN─────────────────────────────────────┤
   └─CANCEL────────────────────────────────────┘

►──┬───────────────────────────────┬───────────────────────────────────────────►◄
   └─MSGROUTEID──(──user ID──)─────┘
```

Required parameters:

session_id

Specifies the name of an XRC session. This can be any name that was previously specified on an XSTART command, except for the word ALL. The maximum length of the name is eight characters. Do not include embedded blanks. TSO ignores leading blanks and trailing blanks.

TIMEOUT

Specifies the minimum time that a storage control can wait for the XRC session to restart and begin to offload records from the primary storage controls. If you do not restart the XRC session within the time that you specified with the TIMEOUT variable, the primary storage controls take action to recover cache resources that are associated with the storage control session. Storage controls that do not support hardware bitmapping *end* their XRC sessions at the end of the specified TIMEOUT duration. Storage controls that support hardware bitmapping *suspend* their XRC sessions and perform hardware bitmapping of changes to primary volumes.

The TIMEOUT parameter allows you to control the impact that a suspension operation has on the application program. The minimum TIMEOUT value is one second (00.00.01); the maximum TIMEOUT value is 18 hours (18.00.00).

The TIMEOUT timeframe begins when update activity causes storage control cache to exceed a threshold value. If an IEA482I message is received, application updates remain stopped until the “busy-control-unit” condition is relieved. If the storage control ends the storage control session, you must restart the storage control session and fully synchronize all volumes in the session.

When you specify TIMEOUT, XRC suspends the entire XRC session, and ends the ANTAS*nnn* system data mover address space. The TIMEOUT parameter is mutually exclusive with the VOLUME parameter.

When you issue the XSUSPEND command with the TIMEOUT(STORAGE_CONTROL_DEFAULT) parameter, XRC uses the default that is set in the storage control. This value is normally five minutes (00.05.00).

Note:

1. If all of the storage subsystems in your environment support the hardware bitmap function, you can set a low timeout value; the hardware bitmap ensures a rapid resynchronization time. If some or none of the storage subsystems in your environment support the hardware bitmap function, set the timeout limit to minimize the effect on applications. Take into consideration how long a full-volume resynchronization will take for storage subsystems without hardware bitmapping.
2. ESS storage subsystems suspend their sessions as a part of XSUSPEND processing. Therefore, the TIMEOUT parameter does not affect these subsystems.
3. The XSUSPEND TIMEOUT parameter specifies the acceptable application impact (expected duration) for a planned outage. For comparison, the XSET TIMEOUT parameter specifies the acceptable application impact for normal SDM operations.

VOLUME

Specifies the volume serial numbers of the primary volumes that XRC is to suspend from the session. When XRC suspends a volume, the SDM continues to read record updates from the primary storage control. The SDM does not apply updates to the secondary volume while the volume remains in a suspended state. You may specify up to 100 primary volume serial numbers on a command. Use either a blank or a comma to separate each primary volume serial number.

The XSUSPEND command will fail if you include a utility volume in the volume list. You can add or delete utility volumes, but you cannot suspend them.

Note: Although you cannot explicitly suspend a utility volume, XRC can internally suspend a utility volume if all of the data volumes in the storage control session are suspended by command or due to an error. When Enhanced Multi-Reader is in effect, all utility volumes in the Enhanced Multi-Reader group for that storage control session are suspended.

When you specify XSUSPEND VOLUME(ALL), XRC suspends all volume pairs and all utility volumes in the session. The ANTAS*nnn* address space remains active. The system data mover continues to read record updates for storage controls that do not support a hardware bitmap.

The VOLUME parameter is mutually exclusive with the TIMEOUT parameter.

For additional information about the potential impact on the application program, refer to "Suspending an XRC session" on page 180.

Optional parameters:

ATTIME

Specifies that XRC suspend the specified volume pairs or the XRC session after all updates, up to the time specified with the timestamp, and apply them to the secondary volumes. XRC does not apply any updates to the secondary volume beyond the specified UTC timestamp.

When XRC encounters a record update that has a timestamp later than the time specified, XRC suspends the volume pair. All affected secondary volumes are consistent up to the UTC timestamp that is reported on successful completion of the command.

If you specify VOLUME(ALL) with the ATTIME parameter, XRC processes volumes that are in duplex, pending, seqcheck, and copy status. The XSUSPEND command does *not* process utility volumes when you specify VOLUME(ALL).

You may issue another XSUSPEND command to change the ATTIME value to be either earlier or later than a previously specified time. XRC issues a message that indicates that it has accepted a new ATTIME value. The command fails if the specified ATTIME value is earlier than the consistency group time that XRC is currently processing. If you issue multiple commands for either a single volume pair or an entire session, XRC always uses the last valid ATTIME value that was specified.

If all volume pairs become deleted or suspended (either due to an error or by command) prior to the specified ATTIME, XRC cancels the XSUSPEND command and clears the ATTIME timestamp.

For *volume* suspension, record updates that are received in the storage control after the specified time are maintained in the software bitmap for a subsequent resynchronization operation. For *session* suspension, storage controls that support hardware bitmapping suspend their storage control sessions and perform hardware bitmapping of changes to primary volumes. The system data mover stops reading record updates from the storage control. If a storage control does not support a hardware bitmap, the storage control accumulates record updates in cache until either you restart the XRC session, or the storage control reaches the timeout value.

If you issue an XRECOVER command to a session that contains suspended volumes pairs, XRC clips those secondary volumes with the primary volume serial numbers, provided that the volume pair had reached duplex prior to being suspended. XRC does not apply journal updates to the volumes. The system data mover ensures that the proper data is recovered from the journal data set so that volumes are consistent to the reported timestamp.

DRAIN

Specifies that XRC reads the record updates from all storage control caches and obtains the timestamp of the last update for each storage control session. Specifying this parameter ensures that XRC applies all updates to the secondary volumes before it suspends either the volume pairs, or the XRC session.

If a storage control has never received a timestamped record, the DRAIN function fails. XRC cancels the command if it suspends the volume pair because of an error prior to completion of the function.

If you specify VOLUME(ALL) with the DRAIN parameter, XRC processes volumes that are in duplex, pending, seqcheck, and copy status. The XSUSPEND command does *not* process utility volumes when you specify VOLUME(ALL).

All secondary volumes are consistent up to the UTC timestamp that is reported by successful completion of the command. The storage control continues to receive record updates after the time that is determined by the XSUSPEND DRAIN function. These updates are maintained in the software or hardware bitmap if this is a volume suspension, or maintained in the primary storage control if this is a session suspension.

If an XRECOVER command is issued, the secondary volumes of volume pairs in the duplex state or the seqcheck state are clipped to the primary volume serial number. XRC does not apply journal updates to the volumes. Although XRC does not apply journal updates to suspended volumes, XRC does clip volume serial numbers to pairs that were in duplex state prior to the suspension. The system data mover ensures that the proper data is recovered from the journal data set so that volumes are consistent to the reported timestamp.

If all volume pairs become deleted or suspended (either due to an error, or by command) before XRC can drain all updates from the primary storage control, XRC cancels the XSUSPEND command.

CANCEL

Specifies that XRC cancel the previous XSUSPEND command with an ATTIME or DRAIN request. The CANCEL option is only valid when there is a pending XSUSPEND command, and cannot be specified with the ATTIME, DRAIN, or IMMEDIATE parameters.

IMMEDIATE
Specifies that XRC immediately process the suspend request. Updates to the specified secondary volumes stop when the current consistency group has been applied. The storage control continues to record updates to primary volumes.

For *volume* suspension, XRC suspends the volume pair or pairs from the active session. For *session* suspension, XRC suspends all volume pairs from the active session. In either case, the secondary volumes are consistent up to the UTC timestamp that is reported on the successful completion of the command. IMMEDIATE is the default.

If you want to process all volumes in the session, except for utility volumes and volumes that are already suspended, specify both the VOLUME(ALL) and IMMEDIATE parameters. The XSUSPEND command does *not* process suspended volumes or utility volumes.

NOVERIFY
Specifies that XRC not issue message ANTT0028A, which prompts the user to confirm that XRC should continue to process the command. If you select this option, XRC immediately processes the command without waiting for verification.

MSGROUTEID
Specifies the user ID to which XRC messages associated with the processing of this command are routed. If the specified user ID is not logged on, TSO saves messages in the TSO BRODCAST data set and displays them after the next logon.

Note: Messages issued by the initial command parser routine are sent to the user ID that issues the TSO command.

XSUSPEND command examples

The following are examples of the XSUSPEND command:

```
XSUSPEND TUCSON CANCEL

XSUSPEND TUCSON TIMEOUT(S) MSGROUTEID(OPER1)

XSUSPEND TUCSON V(prod14) ATTIME(1997.234 12:14:20.002001)

XSUSPEND TUCSON V(prod15,prod16) CANCEL
```

XSUSPEND in a coupled environment

The following section provides information about:

- Suspending all coupled sessions in a master session
- Suspending volumes in all coupled sessions associated with the master session

XSUSPEND of all coupled XRC sessions: In a coupled environment, you can issue the XSUSPEND command to a master session. All sessions are suspended with their active duplex volumes at a consistent time across the coupled sessions.

You can issue the XSUSPEND command to an individual XRC session only:

- After the XRC session has been uncoupled
- If the XRC session is in a coupled state with all volumes suspended
- If there are no volumes in the XRC session

If you issue an XSUSPEND command with TIMEOUT(*hh.mm.ss*) to an XRC session that is part of a coupled session, the command is rejected. A return code is returned that indicates that the session is coupled.

To uncouple and suspend an XRC session that is part of a coupled session, you can use one of the following methods:

- Issue an XSUSPEND command to the master session. The session will be consistent to the same time as the other sessions in the master session.
- Suspend or delete all volumes in the XRC session and then issue an XSUSPEND command to the XRC session. Or issue an XCOUPLE command with the DELETE option and then issue the XSUSPEND command to the XRC session. In either case, the session will no longer be consistent to the same time with the other sessions in the master session. As a result, you will not be able to recover all volumes to a consistent time with the other volumes in the master session.

If you issue an XSUSPEND command for one or more volume pairs in a session, the command operates in the same manner in a coupled or noncoupled environment. The system suspends the specified volume or set of volumes and manages the remainder normally. If the last volume in a coupled session is suspended, the status of the session changes to NON-INTERLOCKED and the session is not kept consistent to the same time as the other coupled sessions.

The IMMEDIATE, ATTIME, and DRAIN parameters allow you to control when the suspension of the master session or an uncoupled XRC session occurs.

After you issue the XSUSPEND command to the master session, XRC prompts you (unless you have specified NOVERIFY) to confirm the request to suspend the master session or set of specified volumes. You must specify either `YES` or `NO`. If the XSUSPEND command is invoked through a CLIST written in REXX, you should either specify NOVERIFY or place your response to the prompt in the data stack prior to issuing the command. In a batch environment, you can stack the response.

XSUSPEND of all volumes in coupled XRC sessions: In a coupled environment, you can suspend the volumes for all the sessions coupled to the master session by issuing the XSUSPEND command to a master session and specifying the VOLUME(ALL) option.

Issuing the XSUSPEND *msession_id* VOLUME(ALL) command causes individual XSUSPEND commands to be issued to each XRC coupled session in the master session, which results in all volumes for the session to be suspended. You cannot specify the CANCEL option with the XSUSPEND command if issued to a master session name.

XSUSPEND in a coupled environment command syntax: The syntax of the XSUSPEND in a coupled environment command is:

```
►►──XSUSPEND──msession_id──┬──────────────────────────┬──────────────────────────►
                           │              ┌─SYS1─┐    │
                           └─MHLQ──(──────┴─mhlq─┴──)─┘
```

```
►──┬─TIMEOUT──(──┬─hh.mm.ss──────────────┬──)─┬──────────────────────────────────►
   │             └─STORAGE_CONTROL_DEFAULT─┘    │
   └─VOLUME──(──ALL──)──────────────────────────┘

   ┌─IMMEDIATE───────────────────────────────┐
►──┼─────────────────────────────────────────┼──┬──────────┬─────────────────────►
   ├─ATTIME──(──yyyy.ddd hh:mm:ss.thmiju──)──┤  └─NOVERIFY─┘
   ├─DRAIN───────────────────────────────────┤
   └─CANCEL──────────────────────────────────┘

►──┬────────────────────────────┬─────────────────────────────────────────────►◄
   └─MSGROUTEID──(──user_ID──)──┘
```

Required parameters*:*

msession_id
: Specifies the name of the master session. The master session is a logical entity that is used to coordinate session commands and data consistency across multiple XRC sessions. A master session exists as long as there is an XRC session coupled to the master session. The maximum length of the name is eight characters. Do not include embedded blanks. TSO ignores the leading blanks and the trailing blanks.

TIMEOUT
: The TIMEOUT(*hh.mm.ss*) parameter may be applied with the XSUSPEND(*msession_id*) command only after a coupled XRC session is uncoupled.

VOLUME
: Uncouple XRC sessions from the master session by issuing the VOLUME(ALL) parameter to the master session. You may also use an abbreviated syntax of V(ALL).

Optional parameters*:*

MHLQ*(mhlq)*
: Specifies the high-level qualifier of the master session control data set. The default is SYS1. If you specify the MHLQ parameter on the XSUSPEND command, and the *session_id* specified is an XRC session name, the MHLQ parameter will be ignored.

ATTIME
: When XSUSPEND is issued using a master session name, the ATTIME parameter is processed as global ATTIME if no component session consistency time has passed the requested timestamp. If any session has passed the time, the command is rejected.

DRAIN
: All sessions in a coupled master session determine a 'drain time'. When XSUSPEND is issued using a master session name, the DRAIN parameter is processed as ATTIME by using the latest component drain time.

CANCEL
: Specifies that XRC cancel the previous master XSUSPEND command that has not yet processed. The CANCEL option cancels a pending XSUSPEND command that was issued to the master session, but not yet processed. If the XSUSPEND command has started processing on any session, the CANCEL request fails.

A pending XSUSPEND VOLUME(ALL) to a master session cannot be canceled from the master session. Instead, you must issue an XSUSPEND CANCEL VOL(*volser_list*) from each individual coupled session to specify all volumes in that session by primary volume serial number.

IMMEDIATE
When you issue the XSUSPEND command by using a master session name, the command will be processed on all XRC sessions that are coupled to the master session. In this instance, the IMMEDIATE parameter is processed as ATTIME equal to a time that permits all coupled sessions to be ended at the same consistency time.

NOVERIFY
Specifies that XRC not issue message ANTT0028A, which prompts the user to confirm that XRC should continue to process the command. If you select this option, XRC immediately processes the command without waiting for verification.

MSGROUTEID
Specifies the user ID to which XRC messages associated with the processing of this command are routed. If the specified user ID is not logged on, TSO saves messages in the TSO BRODCAST data set and displays them after the next logon.

Note: Messages issued by the initial command parser routine are sent to the user ID that issues the TSO command.

XSUSPEND in a coupled environment command examples: The following are examples of the XSUSPEND in a coupled environment command:

```
XSUSPEND MSTR3 TIMEOUT(S) MSGROUTEID(OPER1)

XSUSPEND MSTR2 V(ALL) ATTIME(2000.234 12:14:20.002001)
```

ANTRQST application programming interface overview

The ANTRQST assembler macro provides an application program call to the z/OS system data mover's application programming interface (API). This macro allows you to call XRC, PPRC, FlashCopy, and SnapShot copy functions. You might need to ensure that necessary address spaces, such as ANTAS000, ANTAS001, or ANTMAIN, are active when you process this macro.

A QFRVOLS request has been added to the ANTRQST macro. DFSMSdss invokes the QFRVOLS request to determine if volumes are eligible to participate in a fast replication operation. In addition, the SCREXTENTS and TGTEXTENTS of the FCESTABLISH and FCWITHDRAW requests allow target extent relocation. The specified target extent does not have to match the corresponding source extent.

For additional information about the ANTRQST assembler macro, refer to Appendix C, “ANTRQST and ANTRQSTL macros – call to the system data mover API,” on page 573.

For additional information about fast replication functions and FlashCopy, refer to Chapter 26, “What is FlashCopy?,” on page 471.

Chapter 6. Administering your extended remote copy environment

This chapter discusses parameter library, patch, and flag support. Parameter libraries (parmlibs) are partitioned data sets in which reside a list of commands and options that can determine an operating environment for a program such as XRC. Parmlib members provide a single method of specifying parameters for each system in a multisystem environment without having the user enter the parameters directly for each system.

Flags are diagnostic tools that perform a specific action if the triggering event occurs. Users turn these flags on and off according to their preferred response to the triggering event.

Patches are maintenance tools that provide information for temporary operation. Patches allow the user to take corrective actions, if necessary.

For additional information about Parameter libraries, refer to *z/OS MVS Initialization and Tuning Reference*.

In this topic

This topic provides an overview of XRC parmlib support and the parameters that you can include in a parmlib member. It includes the following sections:

Description and use of parameter libraries (Parmlibs)

The use of parmlibs allows you to tailor operations to your own installation requirements, which improves the performance of XRC. Most of the XRC control values that parmlib can alter are centralized in a tuning table.

With parmlib support, you can modify the values of the XRC tuning table by using the XSET command or parmlib parameters. In general, you should not change these values. The default values set for certain XRC parameters work for most customer configurations. However, through experience or testing, some users have found that they can help improve XRC performance by increasing or decreasing the default values.

With parmlib support, you can set up your environment before starting XRC. This makes it easier to track changes and offers more control over your system. Afterwards, you can dynamically override some parmlib parameters by issuing the XSET command. The corresponding XSET parameters are included in parentheses next to the associated parmlib parameter.

Recommendation: Exercise caution when changing the default values associated with the tuning table. It is recommended that you contact an IBM representative before you attempt any change to the tuning table values.

PARMLIB members

Parmlib concatenation allows concatenation of up to 10 libraries to data set SYS1.PARMLIB. It also provides message IEE252I in syslog to show in which data set a member resides.

When XRC is started by the XSTART command, XRC looks for member ANTXIN00 in the parmlib concatenation. Any of the parameters that are described in Table 22 on page 127 can be included. The parameters that are unique to ANTXIN00 are processed first so that XRC knows which libraries and members to use to finish parmlib processing. These parameters (*hlq*, *mhlq*, *parmlib*, *global* and *session*) are found in the NAMES and STARTUP categories.

It is possible to change the parameters in the NAMES and STARTUP categories in the ANTXIN00 member. The following parameters describe some of the changes that you might want to make:

- Parameter *hlq* provides the high-level qualifier for all XRC data sets. The default is SYS1, but you can override this value with the *hlq* parameter that is specified on the XSTART command.
- Parameter *parmlib* allows you to specify which parmlib to use for the processing that follows. If XCOPY is specified (the default), data set *hlq*.XCOPY.PARMLIB is used. If you specify SYS1, the parmlib concatenation is used. If the parmlib concatenation is used, any messages that include the parmlib data set name display SYS1.PARMLIB+.
- Parameter *global* allows you to specify a single member containing XRC parmlib values that are applied to all XRC sessions.
- Parameter *session* allows you to specify a single member for each XRC session. Each member contains XRC parmlib values which are applied to the associated XRC session.

When you issue the XSTART command, XRC performs the following:

- Searches for member ANTXIN00 in the parmlib concatenation. If ANTXIN00 is not found, the defaults of *hlq*(SYS1), *mhlq*(SYS1), *parmlib*(XCOPY), and *global*(ALL) are used. In addition, the default member used for a session is the member having the same name as the *session_id*.
- Processes the parameters found in ANTXIN00. XRC processes the parameters for *hlq*, *mhlq*, *parmlib*, *global*, and *session*. These parameters are processed only if they are found in ANTXIN00 during XRC XSTART processing. You can choose to not have any other XRC parmlib members, in which case you might provide additional XRC parmlib values in the ANTXIN00 member; these additional parameters are also processed at this time.
- Searches for the member specified on the *global* parameter. XRC parmlib values specified in this member are applied to all XRC sessions. If the Global parameter member was not specified , member ALL is searched for, and if found, XRC parmlib values in this member are processed.
- Searches for the member specified on the *session* parameter for the session being started. The Session parameter might contain multiple pairs of names: the first name in a pair is the *session_id* name and the second name in the pair is the member name containing the XRC parmlib values that will be used for the session. If a matching *session_id* name is found, the associated member is

searched for. If the associated member is found, the parmlib values specified in this member are processed. If the *session_id* is not in the Session parameter, the member with the name of *session_id* is searched, and if found, XRC parmlib values are processed from this member.

The general format of the SESSION parameter is as follows:

```
Session(session_id1 member1 session_id2 member2)
```

The following example represents approved coding for the Session parameter:

In this example, the *session_id* of started sessions DBACKS and ANGELS is specified on the Session parameter, and XRC processes parmlib member BIGUNIT for session DBACKS and parmlib member SMALUNIT for session ANGELS.

```
SESSION(DBACKS BIGUNIT ANGELS SMALUNIT)
```

NOTE: If the same parameter value appears more than once in a parmlib member or in more that one parmlib member, XRC uses the value it processes last. For the Session parameter, if a *session_id* is specified more than once, only the XRC parmlib values for the last member found is applied.

It is possible for errors to occur as the parameters process. If XRC finds an error, it issues an error message to the operator console describing the error. When ANTAS000 starts up and errors are found, the system issues message ANTI1030E and processing continues. When an ANTAS00*n* address space starts up because of an XSTART command and errors are present, the processing of the XSTART command ends with return code 4119.

When an XRECOVER command is issued, only parameters specified in the ANTXIN00 member are processed.

For information about the allocation specifications of *hlq*.XCOPY.PARMLIB, refer to "Creating a PARMLIB data set" on page 62.

For additional information about parmlib concatenation, refer to *z/OS MVS Initialization and Tuning Reference*.

Updating the ANTXIN00 member

Perform the following steps to make changes to the ANTXIN00 member:

1. Update the ANTXIN00 member.

2. Stop any applications that issue TSO commands or XRC API requests through the ANTRQST macro.

3. Cancel the ANTAS000 address space.

Canceling the ANTAS000 address space causes it to reinitialize. Initialization must complete before any applications can issue commands or requests. If you do not stop these applications, the TSO commands and ANTRQST requests might fail. Certain commands (for example, XQUERY MASTER) execute in address space

ANTAS000. If you do not cancel ANTAS000 after updating ANTXIN00, the commands which executed in address space ANTAS000 will not incorporate the changes.

Authorization

If it is necessary to authorize XRC to access the parmlib concatenation, use the following procedure:

1. Associate the XRC address spaces with an RACF user ID. The best way to do this is to define a generic profile in the RACF STARTED class, using the following RACF command:

   ```
   RDEFINE STARTED ANTAS00%.* STDATA(USER(ANTSTC) GROUP(ANTOPER))
   ```

   Note:

   a. The RACF command defines the ANTAS00*n* procedure to RACF.
   b. This example assumes that you have already defined the ANTSTC user ID , and that ANTSTC is associated to group ANTOPER.
   c. Either the ANTSTC user ID or the ANTOPER group must have READ access to all data sets in the parmlib concatenation.

2. If STARTED profiles are RACLISTed on your system, you must refresh them to make the change issued in Step 1 effective. To do this, issue the following RACF command:

   ```
   SETROPTS RACLIST(STARTED) REFRESH
   ```

XSET PARMLIB

To invoke parmlib support at times other than XRC start up, you can issue the XSET PARMLIB command. You can use the XSET PARMLIB both before and after you issue an XSTART command. If you invoke XSET PARMLIB before an XSTART command, you can check the validity of the parameter syntax without applying any of the parameters. Syntax checking looks for occurrences of data specification errors, such as missing commas and illegal parameters, as well as the validity of the commands and their structure. If you invoke XSET PARMLIB after an XSTART command, you can apply the command parameters as well as perform syntax checking. The following statement shows the XSET PARMLIB command syntax:

```
XSET session_id PARMLIB( member_name ACTION(action) DATASET(parmlib_dataset_name) )
```

For additional information about the XSET PARMLIB command, refer to "PARMLIB" on page 104.

ANTXIN00 parmlib parameters

Parmlib parameters use syntax similar to that of TSO commands. Parameters consist of a category name, parameter names, and values. Table 22 on page 127 provides a list of the possible parmlib parameters, as well as their associated XSET parameter, if applicable. XRC references static (S) parameters only once when the corresponding function is first invoked. XRC references dynamic (D) parameters continually each time the function is executed. The rightmost column lists the tuning tables' offset value that is associated with the parmlib parameter.

You can continue parameters from one line to the next by placing a dash (-) as the last character on the line. You can add comments with the /* */ syntax. The following rules apply to comments:

- Comments cannot span multiple lines. If a comment is longer than a single line, break it into multiple lines that contain an opening /* and a closing */ on each line.
- Comments must be to the left of any continuation sign. The dash that indicates a continuation in the command must be located to the right of any comments for that line.
- You must include a continuation character on the comment line when the comment line is placed in the middle of lines that are being continued.

For an example of the correct parmlib syntax, see “PARMLIB example” on page 154.

To determine the current settings for the parmlib parameter values, issue the XQUERY ENVIRONMENT(PARM) command. To find the default values for the parameters, issue the XQUERY ENVIRONMENT(PARM) command to an inactive or undefined session. The values displayed are from the control address space (ANTAS000). There are example output screens in “Examples of XQUERY ENVIRONMENT reports” on page 155. See “XQUERY–Querying a session” on page 90 for more information on viewing environment settings.

Table 22. ANTXIN00 parmlib parameters

Parmlib parameter	Associated XSET parameter	Ranges/values	Dynamic / Static	Default Value
Category: BITMAP				
ChangedTracks	RTRACKS	0–99999	D	7500
DelayTime	RFREQUENCY	00.00.00, 00.00.30–18.00.00	D	00.30.00
Category: CONTIME				
DefaultSessionID	-	-	D	-
DefaultHlq	-	-	D	SYS1
Category: COUPLING				
DatasetDelay		25–250	D	45
DeadSessionDelay		10–120	D	45
Category: DIAG				
SCDumpType		STATESAV, NDSS	S	STATESAV
Category: IOTIMING				
InitializationReadWrite		0–255	D	120
MinExtenderRead		0–255	D	55
MinLocalRead		0–255	D	0
MiscHigh		0–255	D	15
MiscLow		0–255	D	2
ShadowRead		0–255	D	10
ShadowWrite		0–255	D	10
ShadowTimeoutPercent		10–90	D	40
Category: MONITOR				

Table 22. ANTXIN00 parmlib parameters (continued)

Parmlib parameter	Associated XSET parameter	Ranges/values	Dynamic / Static	Default Value
MonitorOutput		ON, OFF	D	OFF
MonitorWakeup		5000–120000	D	10000
Category: NAMES				
Hlq		–	S	SYS1
MHlq		–	S	SYS1
Category: SHADOW				
AllowEhancedReader		Yes, No	D	No
ConsistencyGroupCombined		1–999	D	5
DeviceBlockingThreshold		0–255	D	20
DfltWritePacingLvl	DVCBLOCK	0-F		0
JournalPriority		251–253	D	251
LowAttention		1–255	S	192
MaxBytesTransferred		0, 60000–9999999	D	512500
MaxTotalReaderTasks		32-80	D	40 if AllowEnhancedReader(NO) 32 if AllowEnhancedReader(YES)
MaxTracksFormatted		0–999	D	0
MaxTracksRead		1–255 Values above 246 are accepted but are equivalent to 246.	D	64
MaxTracksUpdated		0–999	D	0
NoTimeStampCount		0–99999	D	5000
NumberReaderTasks		Tuples containing (SCSN, #tasks) #tasks range is 0-16	D	(*, 0) Use the number of XRCUTL volumes.
PacingReportThreshold		0-255	D	10
PavByteThreshold		60000–9999999	D	512500
PavVolumes		1–9	D	1
ReadDelay		100–5000	S	1000
ReaderPacingLimit		20–65	D	33
ReaderPacingWindow		1–30	D	3
ReadRecordsPriority		251–253	D	252
RequireUtility		YES, NO		YES
ResidualLeftToRead		1–500	D	128
ScheduleVerify		YES, NO		NO
StorageControlTimeout	TIMEOUT	00.00.00–18.00.00	D	DEFAULT
SuspendOnLongBusy		YES, NO		NO

Table 22. ANTXIN00 parmlib parameters (continued)

Parmlib parameter	Associated XSET parameter	Ranges/values	Dynamic / Static	Default Value
UtilityDevice	UTILITY	FLOAT, FIX	D	FIX
VerifyInterval		0–24		24
WriteRecordsPriority		251–253	D	253
WrtPacingResidualCnt		0–255		80
WorkloadWritePacing		DISABLED or 6 values 0-F	Initial S, Change D	None
Category: STARTUP				
ClusterMSession		lists of system name and master session name		DISABLED
ClusterName		lists of system name and cluster session name	D	*
Global		*member_name*	S	
Hlq		String for high level qualifier (8 bytes max)	S	SYS1
MaxControlTasks		128–233		128
MHlq			S	SYS1
OfflineDiscovery		YES, NO	S	NO
Parmlib		XCOPY, SYS1	S	
Session		lists of *session_id member_name*	S	
SuppressTimestamp		YES, NO	S	NO
zIIPEnable		FULL, YES, NO	D (ANTAS0*nn* only)	NO
Category: STORAGE				
BuffersPerStorageControl		100–25000	D	576
PermanentFixedPages	PAGEFIX	0–9999	D	8
ReleaseFixedPages		YES, NO	D	NO
TotalBuffers		100 - 25000	D	25000
IODataAreas		100–9999	D	256
Category: VOLINIT				
EnableRefreshs	REFRESHS	YES, NO	D	NO
InitializationMethod	COPY	FULL, QUICK	D	FULL
InitializationsPerPrimary	SCSYNCH	0–45	D	2
InitializationsPerSecondary	SCSYNCH	0–45	D	2
HaltAnyInit		YES, NO	D	NO
HaltThreshold		0–65535	D	5120
MaxNumberInitializations	SYNCH	0–45	D	4

Table 22. ANTXIN00 parmlib parameters (continued)

Parmlib parameter	Associated XSET parameter	Ranges/values	Dynamic / Static	Default Value
SelectionAlgorithm	PRIORITY	LOAD, FIFO, SIZE	D	LOAD
TracksPerRead		1–15	D	3
TracksPerWrite		1–15	D	3
SecondaryDeviceRange		comma or space delimited ranges	D	NONE
SecondaryVolserPattern		comma or space delimited patterns	D	NONE

BITMAP

XRC maintains two resynchronization bitmaps for each primary volume in a session, called *n* and *n-1*. The active bitmap is *n*, and the inactive bitmap is *n-1*. As an application program writes to a primary track, XRC records the change made by turning on a bit in the *n* bitmap. Periodically, the two bitmaps are switched in order to reduce the number of tracks that have to be copied in a recovery situation. The following parameters determine when the switching occurs.

ChangedTracks (RTRACKS)

Specifies the number of tracks that must change before the system data mover (SDM) switches the resynchronization bitmaps. You can specify a value from 0 to 99999. XRC does not use the number of changed tracks to determine whether to switch the bitmaps when the following conditions exist:

- The tracks value is set to zero
- The tracks value is set to a value that is greater than the number of tracks on a volume

DelayTime (RFREQUENCY)

Specifies how long the system data mover waits before it switches the resynchronization bitmaps. You can specify the frequency in hours, minutes, and seconds. You can specify a value between 00.00.30 (30 seconds) and 18.00.00 (18 hours). If you set the frequency value to zero (00.00.00), XRC does not use elapsed time to determine whether to switch the bitmaps.

Note:

1. The ChangedTracks and DelayTime parameters control how often the system data mover switches the resynchronization bitmaps. These parameters are triggers for the switch process based on either changed tracks or elapsed time. When the switch process is triggered, both time and track counters are reset. Each volume's bitmap is reset individually and independently of other volumes starting from the time when the volume first started synchronization. This process tends to occur randomly for a number of volumes in any given interval.
2. In general, the defaults that are set for the ChangedTracks and DelayTime parameters work for most configurations. However, there might be benefits in decreasing these parameter values to minimize the amount of data that is transferred after returning from suspension. For instance, you would want to decrease the parameter values if:

- You are running in an environment with a limited network configuration (and slower synchronization).
- You are running in an environment with a high probability of short outages on the SDM connectivity.

3. When you set the ChangedTracks and DelayTime parameters, ensure that the settings do not refresh the volume bitmaps more than once every 5–10 minutes. For example, if the bitmaps are cleared at one minute intervals, significant subsystem and processor resources are used. Setting a higher value than the default for the ChangedTracks and DelayTime parameters can result in a longer volume resynchronization time. This is because it is likely that more data has changed on the primary volumes. Conversely, setting a lower value for these parameters can put a greater demand on the system data mover MIPS (for software bitmaps) or disk subsystem resources (for hardware bitmaps).
4. In an extremely large and active configuration, you might consider increasing the ChangedTracks value to a number greater than 7500; otherwise, switching might be performed too frequently. Also, when you are using a majority or mostly 3390-9 devices, the ChangedTracks option is more dominant.
5. For an extended outage, it is likely that the number of tracks updated during an outage are greater than those in the bitmap at the time of the outage. Therefore, the ChangedTracks and DelayTime parameters have less effect than for short outages. You must consider the expected length of an XRC outage when you evaluate the benefit that is gained from changing these parameter values versus the increased overhead.

CONTIME

This category of parameters is used by the ANTRQST API request of ILK=XRC XCONTIME to provide the session ID and High Level Qualifier values for accessing the dataset containing the session consistency time of volumes used by z/OS System Logger.

Note: If the *session_id* is specified as ANTAS000 in the XSET command, the parmlib values are applied to the ANTAS000 address space and not to a specific session.

DefaultSessionID

Specifies the session id to be used in determining the master recoverable time for an XRC master session, or if a master session is not found, the consistency time for an XRC session through the XCONTIME API request. The session id must be the name specified on the MSESSION parameter of the XCOUPLE ADD request, or the session ID of a session specified on the XSTART command.

If an XRC session is coupled to a master session, specify the master session name to get the master recoverable time for the XRC session. If you specify the session id for a coupled session, the consistency time returned will be the data consistency time of the session if XRECOVER has not been executed against the XRC session.

DefaultHlq

Specifies the high level qualifier to be used in accessing the XRC master data set, or the state data set, to acquire the XRC consistency time through the XCONTIME API request. The hlq must be the hlq specified with the MSESSION parameter of the XCOUPLE ADD request (or default) or the hlq specified on the XSTART (or default) with the indicated session name on the DefaultSessionID parameter.

COUPLING
This category of parameters controls the coupling of multiple XRC sessions.

DatasetDelay
Specifies how often XRC reads from the master data set in a CXRC environment. If the XRC sessions are pacing each other, decrease this value. Based on the update activity level of the system, high-stress environments require a lower value than moderate-stress environments. You can set this value between 25 milliseconds and 250 milliseconds.

DeadSessionDelay
Specifies the time (in seconds) an active XRC session or set of active XRC sessions waits for a nonresponsive session, before suspending a session. This parameter applies to a CXRC environment. You can set a value between 10 and 120 seconds.

Some customers have found that increasing this value prevents a premature suspension of the session if an extended event occurs that can obstruct a single XRC session. If you are generating a dump of the XRC session, set this value to a higher value, such as 90 seconds.

DIAG
This category of parameters is used for diagnostics.

SCDumpType
Specifies the type of storage control dump to be taken for errors that result in state saves. Specify STATESAV for a storage control dump with warmstart (the default). Specify NDSS to request that a non-disruptive state save be taken, if the necessary level of microcode is present on the target controller.

IOTIMING
Normally, I/O is not timed. If it has not completed in an installation-defined amount of time (the missing interrupt handler (MIH) value), a warning message is issued to the operator console. This message states that a device interrupt might have been missed for the device. MVS periodically issues this message until either the I/O completes or is canceled. Because XRC is designed to monitor time on all of its I/O, it can tell the operating system it must cancel the operation. This occurs if an I/O operation has not completed in a specified amount of time.

The parameters that make up the IOTIMING XRC category control the timeout values for the XRC I/Os. If the XRC session initiates an I/O that fails to complete in the time specified in these fields, the I/Os end with a permanent error. A value of 0 in any of the following fields specifies no timing. If timing is used, the MIH is ignored for XRC I/Os.

InitializationReadWrite
Specifies the timeout values for the functions that read and write tracks during volume initialization. You can specify a value between 0 and 255. A value of 0 means that no timing is used.

MinExtenderRead
Specifies the minimum number of seconds that are used to time the I/O if XRC is reading record sets through channel extenders. You can specify a value between 0 and 255, where 0 indicates that the parameter must not be used. XRC ignores this parameter unless you set ShadowRead to 1.

MinLocalRead
Specifies the minimum number of seconds that are used to time the I/O if XRC is reading record sets locally. You can specify a value between 0 and

255. A value of 0 indicates that the parameter must not be used. XRC ignores this parameter unless you set ShadowRead to 1.

MiscHigh
Specifies the timeout value, in seconds, for volume initialization that performs the read of the hardware bitmap as well as other functions. You can specify a value between 0 and 255, where 0 indicates that the parameter must not be used. You may want to increase this value in a channel extender environment.

MiscLow
Specifies the timeout value, in seconds, for XQUERY I/O, among other things. You can specify a value between 0 and 255, where 0 indicates that the parameter must not be used. In a channel extender environment, increase this value to 10 seconds.

ShadowRead
Controls the read record set I/O time. The default value directs XRC to calculate the appropriate time based on the type of connection and the XSET TIMEOUT value specified. You can specify a value between 0 and 255 for the exact number of seconds to wait for the I/O to complete. If you specify 0, XRC does not use timing. If the value is set to 1, XRC uses the following method to determine the actual timeout value:

1. Find the minimum timeout value of all the storage disk subsystems. (See StorageControlTimeout under the SHADOW parameter.)
2. Find the maximum value from among the calculation from step 1, MinExtenderRead, and MinLocalRead.
3. Multiply the answer from step 2 by the ShadowTimeoutPercent value to get the number of seconds to wait.

If the I/O times out, XRC performs a test to determine if it is experiencing performance impacts—to determine if XRC is running out of data buffers, for example. If the answer is no, the I/O is issued again. If the answer is yes, XRC returns an error for the I/O.

ShadowWrite
Specifies the timeout value for writing updates to secondary volumes. A value of 0 indicates that the parameter is not used. You can specify a value between 0 to 255 seconds. If the value is 1, XRC uses the following method to determine the actual timeout value:

1. Find the minimum timeout value of all the storage disk subsystems. (See StorageControlTimeout under the SHADOW parameter.)
2. Multiply the answer from part 1 by the ShadowTimeoutPercent value to get the number of seconds to wait.

If the I/O times out, XRC performs a test to determine if it is experiencing performance impacts—to determine if XRC is running out of data buffers, for example. If the answer is no, the I/O is issued again. If the answer is yes, XRC returns an error for the I/O.

ShadowTimeoutPercent
Ensures that I/O times out before a storage control session timeout value is reached. This parameter specifies the percentage to use when XRC calculates read or write record set timeout values. (See the ShadowRead and ShadowWrite parameters.) You can specify a value between 10 and 90 percent.

MONITOR
XRC has a monitor task that periodically checks for delays in processing. If

there are any delays, XRC issues message ANTX8117I. In addition, XRC can save the statistics that are used in making these determinations. XRC uses the value specified with the MonitorOutput parameter to tell if these statistics must be saved.

MonitorOutput
Specifies if the results of a monitor check must be saved. A value of ON routes the results to the MONITOR1 member of the state data set. A value of OFF directs XRC to not write the statistics.

MonitorWakeup
Specifies the delay time between collection of monitor statistics intervals in milliseconds. The default value is 10000 milliseconds (10 seconds). The range of valid values is between 5000 and 120000 milliseconds (5 seconds to 120 seconds).

If XRC sessions are coupled to a master session through a cluster, the MonitorWakeup value must be equivalent to the value specified for the cluster. If not, a message is issued warning that the cluster value is used for the XRC session.

NAMES
Values that are found in this category are only used when the XSTART command processes. To take effect, they must be in member ANTXIN00. If these values are found in any other member, they are ignored.

Hlq
Specifies the high-level qualifier for XRC data sets. The default is SYS1. This parameter is not applied unless it is in the ANTXIN00 member and is processed when the XSTART command processes. It can be temporarily overridden on the XADVANCE, XCOUPLE, and XRECOVER commands. It is permanently overridden on the XSTART command.

MHlq
Specifies the high-level qualifier for the XRC master data set. The default is SYS1. This parameter is not applied unless it is in the ANTXIN00 member and is processed at XSTART time. It can be temporarily overridden on the XCOUPLE, XEND, XQUERY, and XSUSPEND commands.

SHADOW
Shadowing (mirroring) is a main function of XRC. XRC takes a copy of data that is changed on a primary volume and writes it out to the corresponding secondary volume. You can specify the following parameters to control this process.

AllowEnhancedReader
Default value: NO

Specifies the use of the enhanced multiple reader function. Associated values are NO (the default) or YES.

- YES — enables the use of the enhanced multiple reader function if the required microcode features are available. Specifying YES allows you to activate the NumberReaderTasks parameter.

  If AllowEnhancedReader(YES) is specified, then to enable the enhanced multiple reader functions, do the following:

  1. Specify UTILITY(FIX) and RequireUtility(YES).
  2. Set the default value of MaxTotalReaderTasks to 32.
- NO — disables the use of the enhanced multiple reader function.

Note: If you do not suspend the volumes, you can not change the value from YES to NO for an active session . To disable the enhanced multiple reader function, see "Disabling the XRC enhanced multiple reader functions" on page 46 for details.

ConsistencyGroupCombined
Default value: 5

Specifies a value that allows XRC to combine multiple consistency groups when data is written to secondary volumes. You can specify a value between 1 and 999. When using the XRC Multiple Reader function, use values of 5 or lower.

A high value can be advantageous if there is a secondary queue build up because combining consistency groups can result in increased parallelism when writing to the secondary volumes. This is beneficial if there are many volumes that are simultaneously being updated by application programs.

DeviceBlockingThreshold
Defines the number of record sets that must be in cache for a particular device before device blocking for that device is activated. The value can range from 0 to 255. A value of 0 instructs XRC to refrain from device blocking. The actual number of record sets used as a threshold value is the value of this parameter multiplied by 64.

DfltWritePacingLvl
Default value: 0

This value specifies the default session write pacing level to be applied to volumes that have explicitly been assigned the DVCBLOCK(WP0) attribute.

The write pacing level specifies the maximum level of write pacing that can be applied to the primary volume. The levels range from 1 (smallest) to F (largest injected pacing).

If nonzero, the value also applies to new volumes on write pacing-capable controllers that are added to the session without an explicit DONOTBLOCK or DVCBLOCK specification.

If you use the default or specify zero, the system enables volumes without DONOTBLOCK or DVCBLOCK specifications for device blocking. Also, if DVCBLOCK(WP0) is specified on the command, a zero value results in an XADDPAIR, or XSET error.

JournalPriority
Changes the dispatching priority for writing record sets to the journal. You can specify a value between 251 and 253. Higher values position the task earlier in the queue.

LowAttention
Specifies a low threshold value to be used by the storage control of the primary volume when XRC adds a volume pair. You can specify a value between 1 and 255 record sets. If the storage control session contains this number of record sets in cache, XRC raises an attention. This informs XRC that there are record sets that need to be read.

MaxBytesTransferred
Restricts the number of bytes that are written to secondary volumes by a single channel program. Generally, the IBM ESS can write an unlimited amount of data in a single channel program. Older technology disk

subsystems may experience secondary performance slowdown when writing large amounts of data in a single chain.

A value of 0 indicates that the amount of data that is written remains unrestricted. You can specify a value between 60000 and 9999999.

MaxTotalReaderTasks
Default value: If AllowEnhancedReader(NO) is specified, the default value is 40. If AllowEnhancedReader(YES) is specified, the default value is 32.

Controls the limit on the number of active LSS and SCID combinations in a XRC session. The value is 32-80. The parameter prevents the lack of buffer and inefficient processing caused by the excessive numbers of reader tasks. The ability to use a higher number is retained for the benefit of installations that might have more than 32 LSS and SCID combinations active in a session, but using higher numbers is not recommended.

If you increase the MaxTotalReaderTasks value, you can add new LSS/SCID combinations in an XRC session when issuing XADDPAIR commands. If the number of the active sessions is greater than or equal to the number of existing LSS/SCID combinations, you can decrease the value in MaxTotalReaderTasks. When the number of active sessions is less than the number of existing LSS and SCID combinations, you cannot decrease the value the MaxTotalReaderTasks. If you issue the XADDPAIR, the new value in the MaxTotalReaderTasks is used.

Note: If you change the value when XRC is inactive, and the new value is smaller than the number of LSS and SCID's in the session, the XSTART command fails. To reduce the value below the amount used by the session, issue an XDELPAIR command to reduce the actual number of LSS/SCID combinations.

MaxTracksFormatted
Restricts the number of tracks being formatted in a single write record set channel program. A 0 value states that there are no restrictions. You can specify a value between zero and 999.

Note: XRC does not format tracks and update tracks in the same channel program.

MaxTracksRead
Specifies the maximum number of record sets that are read in a single channel program. The maximum number of record sets read in a single channel program is 246. For compatibility with previous releases, MaxTracksRead values between 1 and 255 are accepted, but any value above 246 is treated as equivalent to 246.

MaxTracksUpdated
Restricts the number of tracks being updated in a single write record set channel program. A zero value indicates that there are no restrictions. You can specify a value between zero and 999.

Note: XRC does not format tracks and update tracks in the same channel program.

NoTimeStampCount
Specifies the waiting period before XRC issues message ANTX8030W. If an application program writes to a primary volume, a copy of the changed data is saved in cache in a record set. At the beginning of a record set is a timestamp that indicates when the record set was created. Under certain

circumstances, the timestamp might be zero. This parameter specifies the number of zero timestamps that are generated before message ANTX8030W is issued. You can specify a value between 0 and 99999. A value of 0 instructs XRC to not issue a message.

NumberReaderTasks

Specifies whether auxiliary readers are to be used and the number of the auxiliary readers. Each entry is a tuple. The first value is the SCSN value that the number of read tasks is set for. The second value is the number of read tasks to be used. The default value is (*,0). The default value means using the number of XRCUTL volumes to control the number of auxiliary readers, and applying the value to all SCSNs.

An asterisk (*) in the SCSN field inidicates that the number of tasks value must be applied. Refer to the following two examples:

- (AA, 8) indicates that you must set the number of read tasks for SCSESSION=AA to a value of eight.
- (*, 6) indicates set the number of read tasks for all SCSESSIONS to a value of six.

To enable the specification of NumberReadTasks, use either of the following ways:

- Issue an XADDPAIR command for a volume. The system uses the specifications of NumberReadTasks for each SCSN within each SSID during the XSTART of a new or inactive XRC session. If the new volumes are not added, the existing number of reader tasks might be increased or decreased by the specifications.
- Use the XSET session_id PARMLIB (member, ACTION(APPLY)) command for an active XRC session. You can activate the NumberReadTasks specification by using the PARMLIB to increase the number of reader. If all volumes in a SCSN on a SSID are suspended, you can only decrease the number of reader tasks.

PacingReportThreshold

Default value: 10 (milliseconds)

This value specifies the number of milliseconds of injected write pacing, per write I/O, that must exist for a write pacing-enabled volume to be flagged as “blocked” in XQUERY reports and XRC monitor data. Valid values are 0-255.

Note: When a write channel program creates several record sets, the maximum value is equal to the maximum value of pacing multiplied by the number of record sets created by the channel program. See “Write pacing” on page 160 for details.

PavByteThreshold

Default value: 512500 (bytes)

Specifies the number of bytes used to determine when write I/O for a secondary device is to be split into multiple tasks. When the total number of secondary bytes in a consistency group for a given secondary device exceeds this value, XRC splits the I/O across the number of tasks specified in the PavVolumes parameter.

PavVolumes

Default value: 1

Specifies the number of parallel write I/O tasks to be used when PavBytesThreshold is exceeded for a given secondary device in a consistency group. Use values above 1 only when Parallel Access Volume aliases are available for the XRC secondary volumes.

ReadDelay
Specifies the frequency that XRC checks to see if there are record sets in cache. You can specify a value between 100 and 5000. The value is in milliseconds with a default of 1000 (one second).

You can attain the following results by changing the XRC parmlib to use a smaller SHADOW READDELAY value, such as 500:

- Achieve more aggressive Recovery Point Objective targets.
- Avoid delays from lightly loaded data movers in large scale configurations.

ReaderPacingLimit
Default value: 33 (percent)

Dynamically reduces the effective reader pacing window after “leading” readers have accumulated updates in the specified percentage of available data mover buffers. You can specify values between 20 and 65 percent. Higher values can increase data mover efficiency, but should be used only in environments where data mover buffers are plentiful. In cases where data mover buffers are tightly constrained, a value of 20 should be used.

ReaderPacingWindow
Default value: 3 (seconds)

Limits the extent to which the data mover can “read ahead” in a given session. Since data mover buffers are limited, “leading” readers that are keeping up with application updates must eventually wait for those “lagging” readers that are not keeping up.

This parameter specifies the number of seconds worth of updates that leading readers may accumulate in data mover buffers while waiting for lagging readers to catch up. The larger the pacing window value, the more efficiently the data mover can operate. You can specify values between 1 and 30 seconds, although values larger than 5 seconds should be used with care and only in situations where sparse updates are occurring and data mover buffers are plentiful. In cases where data mover buffers are tightly constrained, a value of 1 should be used.

Note that the use of an excessively large pacing window can lead to a data mover hang, allowing record sets to accumulate in cache with a potentially serious impact to Application I/O.

ReadRecordsPriority
Changes the dispatching priority for read record sets. You can specify a value between 251 and 253. Higher values position the task earlier in the queue.

RequireUtility
Default value: Yes

This parameter optionally prevents XADDPAIR of a non-XRCUTL when an active storage control session does not exist for the specified (or default) SCSESSION. Associated values are NO or YES (the default).

- YES — prevents XADDPAIR of a non-XRCUTL volume pair when an active storage control session does not exist for the specified (or default) SCSESSION.

- NO — enables XADDPAIR of a non-XRCUTL volume to proceed when an active storage control session does not exist.

ResidualLeftToRead
: Specifies the smallest number of remaining record sets that causes XRC to perform another read. You can specify a value between 1 and 500 record sets.

  Example: If you set the ResidualLeftToRead value to 50, and after reading record sets, there are only 40 record sets remaining, XRC does not read them. However, if you set the ResidualLeftToRead value to 50, and there are 50 or more record sets, then XRC issues another channel program to read the remaining record sets.

ScheduleVerify
: Controls whether the data mover periodically schedules mirror status verification. The default value is NO.

StorageControlTimeout (TIMEOUT)
: Specifies the primary storage control timeout value for application impact. The initial value is set when a storage control is installed. If XRC stops reading the session's record sets, after the timeout value has elapsed, the session automatically terminates (for non-2105 disk subsystems) or suspends (2105 disk subsystems). You can specify a value in *hh.mm.ss* format between 00.00.00 and 18.00.00, where 00.00.00 uses the default value set during installation. When a storage control session initially starts, XRC uses the timeout value specified in the parameter. However, you can dynamically alter it using an XSET command. For more information about this option, see "Using the XSET TIMEOUT parameter" on page 183.

  Non-2105 storage controls end their affected storage control sessions at the end of the specified TIMEOUT duration. XRC must then resynchronize all volumes in these sessions with a full-volume copy. 2105 storage controls suspend their affected storage control sessions and perform hardware bitmapping of changes to primary volumes. XRC does not need to perform a full-volume copy to resynchronize these volumes.

  You can override the value specified with this parameter by issuing a timeout value with the XSUSPEND TIMEOUT command. The new timeout value remains in effect until you restart the XRC session.

SuspendOnLongBusy
: Used to enable or disable Suspend on Long Busy function when a new storage control session is added.

  If the storage control microcode supports the function, enabling will cause the microcode to automatically suspend the storage control session instead of raising extended long busy when sidefile limits are exceeded.

  If the storage control microcode does not support the function, enabling will cause the data mover to suspend a storage control session as soon as it detects extended long busy due to sidefile exceeding limits. Storage control sessions that do not support suspension will be terminated.

  Disabling allows the data mover to tolerate the long busy condition for 80% of the storage control session timeout interval, after which mirroring is suspended.

Note: Changing this value does not affect the attributes of existing storage control sessions. Use XSET with the SSID and SUSLBUSY keywords to change the Suspend on Long Busy attribute of existing storage control sessions.

UtilityDevice (UTILITY)
Specifies the method used for selecting a utility device. When you specify FIX, XRC uses the same primary volume. When you specify FLOAT, XRC dynamically picks the primary volume that has the lowest I/O activity.

VerifyInterval
Specifies the interval, in hours, between the scheduled verifications. The default value is 24 hours. A value of zero specifies that verification is performed during every monitor interval.

Note: Continual verification can degrade data mover performance. In production XRC environments, avoid specifying zero.

WriteRecordsPriority
Changes the dispatching priority for writing record sets to a secondary volume. You can specify a value between 251 and 253. Higher values position the task earlier in the queue.

WrtPacingResidualCnt
Default value: 80

This value is multiplied by 64 to determine a target device residual count, at which the maximum permissible pacing will be injected for a given write pacing-enabled volume. A value of 0 has the effect of disabling Write Pacing for a volume when it is next processed with XADDPAIR or XSET DVCBLOCK. Valid values are 0-255.

In general, larger values for this parameter will tend to give higher session delay times and less frequent pacing. Smaller values will tend to give lower session delay times and more frequent pacing.

WorkloadWritePacing
Default value: None

The values are as follows:

DISABLED
Workload-based write pacing should not be used

max-levels
For workload-based write pacing, specifies the maximum write pacing levels that can be applied to application system writes depending on the Workload Manager settings for the application's service class. *max-levels* consists of 6 values, 0-F, separated by commas or spaces. The values correspond to the available WLM settings for the Importance parameter, which has 5 levels and 1 discretionary value (the sixth number). The first value corresponds to WLM importance 1, the second value to importance 2, and so on. The values for WorkloadWritePacing must follow an ascending sequence, that is, each of the second through sixth values must be greater than or equal to the preceding entry, to ensure that high priority workloads (as indicated by a lower value for importance) are paced at a lower level than low priority workloads. Values that are equal to the preceding value in the list are accepted and act as a placeholder for importance levels that are unused at the application site. At least one of the 6 values must be non-zero.

If WorkloadWritePacing is not specified at XSTART, there is no default value. Instead, the WorkloadWritePacing values in effect before the session was suspended are used, or if WorkloadWritePacing was previously disabled, the value from volume-level write pacing parameters that were set on the XADDPAIR or XSET command are used.

To disable workload-based write pacing, you must specify WorkloadWritePacing(DISABLED). Removing the WorkloadWritePacing parameter does not cause XRC to revert to volume-level write pacing when the session is restarted with XSTART, or when you issue the XSET PARMLIB APPLY command. You can use the XSET PARMLIB APPLY command only to change the existing values for *max-levels*.

WorkloadWritePacing with *max-levels* cannot be specified with a value of 0 for WrtPacingResidualCnt.

For more information on write pacing levels, refer to the description of parameter DfltWritePacingLvl in this topic, and the DVCBLOCK parameter in "XADDPAIR–Adding volume pairs or utility volumes" on page 71.

For more information on workload-based write pacing, refer to "Workload-based write pacing" on page 161.

STARTUP

Use the values that are found in this category during the following functions:

- When you issue the XSTART command.
- When you start the MVS MODIFY operation of CREFRESH.
- During startup or restart of the XRC control address space (ANTAS000).
- When you issue the XADVANCE or XRECOVER command to start a cluster session.

Note:

1. If both NAMES and STARTUP are specified in ANTXIN00, and both have either 'HLQ' or 'MHLQ' parameters, then XRC uses the parameters specified under STARTUP.
2. You can change the parameters specified in the STARTUP category and then use the MVS MODIFY operation of CREFRESH. See Appendix A, "Advanced Copy Services diagnostic aids," on page 549 for the syntax of this operation.

ClusterMSession

This parameter allows you to specify the XRC master session name to be associated with a cluster session. The msession id name is the logical session name used on the XCOUPLE command. A value of DISABLED (default) indicates that a cluster session is disabled for a logical partition. If something other than DISABLED is specified, all coupled XRC sessions in a logical partition are coupled to this specified master session through the cluster session.

To specify the XRC master session name, use the following guidelines:

- When you specify a single parameter, use the parameter name as the master session name on the logical partition to access to this parmlib.
- When you specify multiple parameters, specify the parameters in pairs. Each pair represents the system name of the LPAR and the master

session name to be used in the LPAR. Use the system name as the first parameter of a pair. Use the master session name as the second parameter of a pair.

Here is an example for lists of names: where **System 1** represents the system name and **msess1** the master session name for the first partition and **System 2** represents the system name and **msess2** the master session name for the second logical partition.

```
ClusterMSession(System1 msess1-
                System2 msess2)
```

ClusterName

This parameter allows you to specify the XRC cluster session name to be used in a logical partition. This parameter is not applied unless it is in the ANTXIN00 member and is processed when the ANTAS000 address space is started or restarted and when the following MVS console command

```
F ANTAS000,CREFRESH
```

is processed. If no value is specified, the system name for the logical partition is used as the cluster session name (default).

To specify the XRC cluster session name, use the following guidelines:

- When you specify a single parameter, use the parameter name as the cluster session name on LPARs to access to this parmlib.
- When you specify multiple parameters, specify the parameters in pairs. Each pair represents an LPAR system name and the cluster session name to be used in the named LPAR. Use the system name as the first parameter of a pair. Use the cluster session name as the second parameter of a pair.

  Here is an example for lists of names: where **System 1** represents the system name and **cluster1** the cluster session name for the first partition and **System 2** represents the system name and **cluster2** the cluster session name for the second logical partition.

```
ClusterName(System1 cluster1-
            System2 cluster2)
```

Global

Specifies the data set member name containing XRC parmlib parameters which are applied to all XRC sessions when started using the XSTART command.

NOTE: XRC parmlib parameters also specified in the member specified on the Session parameter override the values specified in this Global member.

Hlq

Specifies the high-level qualifier for XRC data sets. The default is SYS1. This parameter is not applied unless it is in the ANTXIN00 member and is processed when the XSTART command processes. It can be temporarily overridden on the XADVANCE, XCOUPLE, and XRECOVER commands. It is permanently overridden on the XSTART command.

MaxControlTasks

This parameter enables you to control the number of tasks available in the ANTAS000 address space for the parallel processing of the following ANTRQST requests:

- ILK=XRC
- ILK=PPRC

The value ranges from 128 to 233. The default value is 128.

Note: GDPS/XRC and GDPS/PPRC installations must have at least one task available for each defined auto-operator.

MHlq
Specifies the high-level qualifier for the XRC master data set. The default is SYS1. This parameter is not applied unless it is in the ANTXIN00 member and is processed at XSTART time. It can be temporarily overridden on the XCOUPLE, XEND, XQUERY, and XSUSPEND commands.

OfflineDiscovery
Controls whether offline device discovery is performed during IPL when the ANTAS000 address space first starts up, and thereafter when ANTAS000 restarts after it has been canceled. A value of YES causes offline discovery to be performed. A value of NO bypasses offline device discovery. NO is the default.

Offline device discovery at IPL should be needed only if you are using one of the following:

- GDPS Metro/zGlobal Mirror (XRC) in HYPER-PPRC mode (Incremental Resynch)
- GDPS Metro/zGlobal Mirror under z/OS V2R1 with XRC offline primary devices.

Other XRC configurations or non-XRC environments do not directly benefit from offline device discovery at IPL. To avoid increasing overall IPL elapsed time unnecessarily, do not request offline device discovery unless it is required.

The value of OfflineDiscovery takes effect when the ANTAS000 address space is started during IPL, or with the automatic restart of ANTAS000 after it has been canceled. To activate a new value for OfflineDiscovery without an IPL, do the following:

1. Update the value in PARMLIB member ANTXIN00
2. Issue the system command CANCEL ANTAS000

The OfflineDiscovery keyword is ignored and is not processed if it is specified in the ALL or session member of *hlq*.XCOPY.PARMLIB.

Parmlib
Specifies which data sets to use when searching for members that contain parmlib parameters. A value of XCOPY (the default) indicates to XRC that it use *hlq*.XCOPY.PARMLIB. A value of SYS1 indicates to XRC that it use the MVS parmlib concatenation. If the parmlib concatenation is being used, messages that include the parameter library data set name display SYS1.PARMLIB+.

Session
This parameter allows you to specify groups of session id name and member name. The session id name is the logical session name used on the XSTART command, and the member name is the data set member name containing the parmlib parameters to be applied to the logical session when the session is started.

Note: The number of parameters specified with Session must be a multiple of 2 (for example, session_id name followed by member name). If multiples of 2 values are not specified, error message ANTI1031E is issued.

SuppressTimestamp

This parameter allows you to suppress channel program timestamping on XRC system data mover systems. A value of YES suppresses timestamping. A value of NO allows timestamping. If SuppressTimestamp is not specified, it is interpreted as SuppressTimestamp(NO).

You should suppress channel program timestamping if the XRC system data mover runs on a system that does not share a common time reference with the application systems that writes to the XRC primary volumes, to avoid the introduction of incorrect timestamps into the XRC storage control sessions.

The recommended use of this parameter is:

- SuppressTimestamp(NO) on application systems with a common time reference
- SuppressTimestamp(YES) on remote systems processing XRC, GDPS control systems (K-systems) and z/OS systems that have access to XRC primary volumes and do not share a common time reference with production systems.

Do not use SuppressTimestamp(YES) on application systems, as that would defeat the data consistency mechanism of XRC. SuppressTimestamp(YES) should be used only on systems that are not updating data or that do not have a common time reference, as a preventative measure to avoid incorrect adjustments to the XRC consistency time.

The value of SuppressTimestamp takes effect when the ANTAS000 address space is started during IPL, or with the automatic restart of ANTAS000 after it has been cancelled. To activate a new value for SuppressTimestamp without an IPL, do the following:

1. Update the value in PARMLIB
2. Issue this system command: `CANCEL ANTAS000`.

The value of SuppressTimestamp takes effect when the ANTAS000 address space is started during IPL, or with the automatic restart of ANTAS000 after it has been cancelled. To activate a new value for SuppressTimestamp without an IPL, do the following:

1. Update the value in PARMLIB
2. Issue this system command: `CANCEL ANTAS000`.

Refer to "Recovery system clock considerations" on page 50 for more information.

zIIPEnable

This parameter specifies whether the ANTAS000, ANTAS0*nn*, and ANTCL0*nn* address spaces are enabled for running on zIIP processors. The values are:

FULL The address spaces are enabled for running on zIIP processors. This option allows the maximum amount of XRC offload possible. The offloaded work will be visible in enclave reports.

YES The address spaces are enabled for running on zIIP processors. This option allows non-I/O related XRC operations to be offloaded to zIIP processors.

NO The address spaces are prevented from running on zIIP processors.

You can change the value dynamically for ANTAS0*nn* by changing the value and using the XSET command to activate the change. The new values apply to all newly created ANTAS0*nn* address spaces.

If you change the zIIPENABLE parameter, restart the XRC address spaces with these commands:

1. Either XSUSPEND TIMEOUT or XEND (according to your local procedures) to end ANTAS0*nn* address spaces
2. XSTART.

zIIPEnable is a global parameter, so the value contained in PARMLIB member ANTXIN00 is used at startup for all address spaces. Note that the zIIPEnable parameter is not processed if it is specified in the global member (specified with a member name of ALL in the global parameter) or in a session member. You can change the parameter dynamically by modifying the value, then using the XSET command to specify its location and activate the change. The updated value is applied to all newly created ANTAS0*nn* and ANTCL0*nn* address spaces and to any existing ANTAS0*nn* and ANTCL0*nn* address spaces that are restarted or that create new tasks.

Changes to this parameter are only recognized by XRC address spaces that are restarted after the change, or for which the XSET command is used to activate PARMLIB changes. The XQUERY ENVIRONMENT(PARM) command shows the current global setting. This setting may differ from the parameter in use by XRC address spaces that were not restarted and were not updated by the XSET command.

STORAGE

XRC uses virtual storage to process customer data. Use this parameter to define the amount of storage that is used in various operations.

The parameters include:

BuffersPerStorageControl

Specifies how many buffers XRC allocates per scsession specified. A value of 576 allocates 35 MB per unique scsession specified with the XADDPAIR command. Some customers have modified this field to a higher number to use the maximum number of buffers, even with a small number of sessions. You can specify a value between 100 and 25000 buffers.

PermanentFixedPages (PAGEFIX)

Specifies the maximum amount of real storage, in MB, that XRC keeps page fixed to process I/O operations. You can specify a value between 0 and 9999 MB. A value of 0 directs XRC to release fixed pages after they have been used.

Example: To minimize the processor load for two storage controls, you would set the PAGEFIX value to 70. The two storage controls divide the 70 MB to get 35 MB each, which is the maximum per storage control session. However, you must allocate 35 MB of storage for each unique scsession level that you initiate. If you have two primary storage controls and start two storage control sessions on each, you would set the PAGEFIX value to 140 MB of storage.

ReleaseFixedPages

Instructs XRC to release fixed pages. XRC tries to free up pages of storage that are not being used. For example, during heavy stress, a large number of buffers are pagefixed to hold the record sets that are being read from cache. Once the heavy demand has decreased, the pages that were fixed are freed if it is determined that the buffers are no longer needed. A value of NO instructs XRC to keep the pages, even if they are no longer needed. A value of YES instructs XRC to release pages that are no longer needed.

Note:

1. If you use ReleaseFixedPages(YES) and need to XDELPAIR all volumes for one or more sessions, IBM recommends that you suspend (throughXSUSPEND TIMEOUT) and restart the data mover session immediately after the XDELPAIR's have completed. Failure to do so can result in extensive storage fragmentation in the data mover address space, resulting in a subsequent inability of the data mover to obtain sufficient contiguous storage for XADDPAIR, XQUERY, and other processes.
2. The number of pages kept fixed for an extended period of time does not exceed the value specified in the PermanentFixedPages parameter.

TotalBuffers

Controls the maximum number of buffers used for an XRC session. Some customers have used a lower value to limit the amount of real storage that is used for an XRC session. You can specify a value between 100 and 25000 buffers.

Lower this value to a level that allows all the buffers to be pagefixed when the system has insufficient real storage to pagefix all of the allocated buffers. Lowering this value might improve performance. It may also reduce processor usage as long as the lower value does not introduce significant performance problems caused by the smaller number of buffers as a working set.

If you apply a PARMLIB change to an active session that decreases the number of buffers available, excess buffers will be freed immediately. This occurs even when RELEASEFIXEDPAGES (NO) is currently in effect. IBM recommends that you make such changes during periods of light workload. Making the change during a heavy workload period may adversely impact session performance.

If you apply a change that increases the number of buffers available, the new buffers are allocated the next time the data mover encounters a data shortage.

Note: This is not an absolute limit on the storage that will be used for buffers in an XRC session.

IODataAreas

Specifies the real storage allocation for XRC channel programs and work areas that are associated with I/O operations. A value of 256 is best for installations that have less than 256 volumes. You can specify a value between 100 and 9999.

VOLINIT

Volume synchronization and resynchronization is the process of copying data from a primary volume to the secondary volume with which it has been associated. The following parameters help control this process.

EnableRefreshs (REFRESHS)

Indicates whether the TSO XSET REFRESHS command is enabled for execution. The default is NO, disabling the REFRESHS command. EnableRefreshs requires a YES value to enable the XSET REFRESHS function. See the REFRESHS parameter description for more information.

InitializationMethod (COPY)
Specifies the extent to which the primary volume is copied to the secondary volume. Specify FULL if the complete primary volume is to be copied to the secondary volume. If only the allocated space on the primary volume is to be copied, specify QUICK. At the beginning of a quick copy operation, XRC performs a reserve against the VTOC of the primary volume to get the allocated extents. After XRC determines the allocated extents, it releases the reserve. For more information about these options, see "Adding a volume with the FULLCOPY or QUICKCOPY option" on page 172.

Note: To ensure data integrity, the initial processing for QUICKCOPY must issue a reserve and then a release for the primary volume. This must be done during the initial phase of the synchronization process. If access to the primary volume is through a channel extender and the connection fails while XRC has the volume reserved, applications at the primary site are not able to access the primary volume.

InitializationsPerPrimary (SCSYNCH *primary*)
Specifies the maximum number of primary volume synchronizations and resynchronizations that can occur concurrently on a single storage control. You can specify a value between 0 and 45. A value of 0 stops the selection process but processing of existing volume synchronization and resynchronization tasks continues.

InitializationsPerSecondary (SCSYNCH *secondary*)
Specifies the maximum number of secondary volume synchronizations and resynchronizations that can occur concurrently on a single storage control. You can specify a value between 0 and 45. A value of 0 stops the selection process but processing of existing volume synchronization and resynchronization tasks continues.

HaltAnyInit
Pauses volume initialization if the record set residual count rises above the HaltThreshold value on any storage control, rather than just the storage control with which the primary volume is associated. A YES value allows pausing of any storage control. A NO value allows pausing for only the primary volume storage control.

HaltThreshold
Stops volume synchronizations or resynchronizations if the record set residual count of the storage control associated with a volume reaches this threshold value. A low value reduces the impact of volume initialization activity that is running concurrently with heavy update activity. If this threshold is reached, volume initialization pauses until the residual count drops below the threshold. When the residual count drops below the HaltThreshold value, the volume synchronizations or resynchronizations begin again. You can specify a value between 0 and 65535. A value of 0 stops all volume synchronizations or resynchronizations regardless of Ioad or IO activity.

MaxNumberInitializations (SYNCH)
Specifies the number of volume synchronizations and resynchronizations that can occur simultaneously in an XRC session. You can specify a value between 0 and 45. A value of 0 stops the selection process, but continues processing existing volume synchronization and resynchronization tasks.

MaxNumberInitializations specifies an XRC session-level value. For each synchronization task that it starts, XRC fixes real page storage that is based on the following formula:

```
{MaxNumberInitializations*(number of pairs in CPY status)*360K}
```

This storage is in addition to the real storage that is used by normal update processing. The real storage remains pagefixed for the total time the volumes are being synchronized or resynchronized.

SelectionAlgorithm (PRIORITY)
Specifies the method the system data mover should use when choosing the next volume pair to be synchronized or resynchronized.

FIFO
Choose volumes in the order that they were added with the XADDPAIR command

LOAD
Choose volumes whose primary storage controls have the least load. XRC considers a primary storage control to be overloaded if it has a higher number of record sets than the value set for the HaltThreshold parameter. LOAD is the default.

XRC bypasses a volume pair if the primary storage control is overloaded. If bypassed, the volume remains the next eligible candidate for XRC to select when the load decreases.

SIZE
Choose volumes based on the total number of cylinders, from largest to smallest. The parameters MaxNumberInitializations, InitializationsPerPrimary and InitializationsPerSecondary are honored; in addition, to avoid overloading storage control sessions, SIZE causes the system data mover to choose the volume with largest number of cylinders on the controller with the least residual load. If there are multiple volumes with the same number of cylinders on controllers with the same residual load, the volume that was XADDed first is processed next.

TracksPerRead
Specifies the number of tracks to be read in a single Read Track channel program. You can specify a value between 1 and 15 tracks.

TracksPerWrite
Specifies the number of tracks to be written in a single Write Track channel program. You can specify a value between 1 and 15 tracks.

SecondaryDeviceRange
Specifies whether a secondary device range filter should be used and if so, the list of device ranges to use. The filter is applied to the secondary device in an XADDPAIR command. If the secondary device number is not within one of the specified ranges, the XADDPAIR command will fail.

Each range contains two four-digit hexadecimal numbers separated by a colon. The value to the left of each colon must be less than or equal to the value to the right of the respective colon. If both values in the range are the same, a filter will be created for the single specified device number. Values with less than four digits are right justified and padded with zeroes. The list of ranges may be separated by commas or blanks. There can be up to 256 ranges in the list.

If SecondaryDeviceRange (NONE) is issued, the list of secondary device ranges will be cleared. If any other device range values are listed along with value 'NONE', the command will fail. Here are a few examples:

- (0F50:0F5F, 0F70:0F7F) means to create secondary device range filters for ranges 0F50 to 0F5F and 0F70 to 0F7F.
- (F5F:F5F F86:F86) means to create secondary device range filters for device number 0F5F and device number 0F86.
- (NONE) means to not use any secondary device range filters.

Once a SecondaryDeviceRange value has been applied, the filter will be used for all subsequent XADDPAIR commands.

The default value is 'NONE'. The default value means that no secondary device range filters will be used when processing an XADDPAIR command.

SecondaryVolserPattern

Specifies whether a secondary VOLSER pattern filter should be used and if so, the list of VOLSER patterns to use. The filter is applied to the secondary device in the XADDPAIR command. If the secondary VOLSER does not match one of the specified VOLSER patterns, the XADDPAIR command will fail.

Each pattern contains a six-character or smaller value with valid characters being those characters which are valid to appear in a VOLSER plus a single-character wildcard '%'. Each pattern must contain at least one single-character wildcard. Lower case is converted to upper case prior to comparison. The list may be separated by commas or blanks. There can be up to 256 patterns in the list.

If SecondaryvolserPattern (NONE) is issued, the list of VOLSER patterns will be cleared. If any other VOLSER values are listed along with value 'NONE', the command will fail. Here are a few examples:

- (X%%%%%) means to create a secondary VOLSER pattern filter for pattern X%%%%%. Acceptable VOLSERs must have an 'X' as the first character and must be one to six characters in length.
- (%%B %) means to create a secondary VOLSER pattern filter for patterns %%B and %. Acceptable VOLSERs must be three characters in length with the third character of 'B', or they must be one character in length.
- (S%S%%%,S%N%%%) means to create a secondary VOLSER pattern filter for patterns S%S%%% and S%N%%%. Acceptable VOLSERs must have an 'S' as the first character, an 'S' or 'N' as the third character, and must be three to six characters in length.
- (NONE) means to not use any secondary VOLSER pattern filter.

Once a SecondaryVolserPattern value has been applied, the filter will be used for all subsequent XADDPAIR commands.

The default value is 'NONE'. The default value means that no secondary VOLSER pattern filters will be used when processing an XADDPAIR command.

Note: If both SecondaryDeviceRange and SecondaryVolserPattern are specified, the device number comparison is performed first with an XADDPAIR command. Therefore if the device number isn't found within any of the specified ranges, the XADDPAIR will fail regardless of whether or not the VOLSER matches a VOLSER pattern.

Description and use of flags

Flags are diagnostic tools that perform a specific action if the triggering event occurs. Users turn these flags on and off according to their preferred response to the triggering event. Examples of the following flags can be found in "PARMLIB example" on page 154.

Flag parameters

The following flag parameters allow you to perform triggered actions in your system. Some of these flags are also listed in Appendix A, "Advanced Copy Services diagnostic aids," on page 549 with the MVS MODIFY commands.

Table 23. Flag Parameters Allow You to Perform Triggered Actions

Category	Name	Action
FLAG		
	ABEND_LIC	ON, OFF
	AUTO_READD	ON, OFF
	IGNORE_INTERRUPTS	ON, OFF
	NRFITF	ON, OFF
	SCTRAP	ON, OFF
	SUPRDUMP	ON, OFF
	TIF_ERROR	ON, OFF
	XENDDUMP	ON, OFF

You can determine the current settings for each indicated flag by initiating the MODIFY command without specifying ON or OFF. To determine the default values for all flag names at once, issue the XQUERY ENVIRONMENT(FLAG) command to an inactive session. This displays the values in the control address space (ANTAS000). For example screens of the XQUERY ENVIRONMENT(FLAG) command, see "Examples of XQUERY ENVIRONMENT reports" on page 155.

Guideline: Although each flag has the ON/OFF option, this does not mean that ON turns the action on and OFF turns the action off. With some flags, the ON option prevents the action from occurring. See the information which follows for a full description of each parameter.

The following values are valid for the Name parameter. Each description includes the function of the Action values for the associated flag name.

ABEND_LIC
: ON causes XRC to request a storage control state save if error message ANTX5106E with return code 638 occurs. OFF prevents XRC from requesting a state save.

AUTO_READD
: ON specifies that if a volume encounters a restartable error, the system data mover automatically re-adds the volume pair to the session when the error suspension is complete. Restartable errors are those associated with unexpected device status in the primary storage subsystem. See the description of return code 904 in *z/OS MVS System Messages, Vol 1 (ABA-AOM)* for more information. The default value is ON. An OFF value disables the function.

IGNORE_INTERRUPTS
ON causes the data mover to ignore low and high attention interrupts which cause record sets to be read from the storage control. The ON option can have the affect of reducing the amount of CPU time used by the data mover.

NRFITF
ON causes XRC to dump storage when it receives return code 901. OFF does not initiate a storage dump with return code 901 type errors; however, an error is still generated.

SCTRAP
ON allows XRC to request that a storage control dump its internal control queues when it encounters certain LIC-related errors. Specifying OFF directs XRC, for certain error types, not to request that a storage control dump its internal control queues. .

SUPRDUMP
OFF directs the system data mover, if required, to dump to a SYS1.DUMP data set when it encounters a software error. Specifying ON suppresses dumps on software-related errors.

TIF_ERROR
ON causes XRC to dump storage when it receives return code 638. OFF does not initiate a storage dump with return code 638 type errors; however, an error is still generated.

XENDDUMP
ON directs XRC to generate a dump whenever a session-level XEND or XSUSPEND command completes. Specifying OFF directs XRC to not generate a dump.

For additional information about:

- XQUERY command, refer to "XQUERY–Querying a session" on page 90.
- XRC SCTRAP operation, refer to "SCTRAP operation (XRC)" on page 558.
- XRC SUPRDUMP operation, refer to "XRCTRAP/SUPRDUMP operation (XRC)" on page 564.
- XRC XENDDUMP operation, refer to "XENDDUMP operation (XRC)" on page 564.

FLAG syntax considerations

Flag names are issued individually. You must issue only one name and one action in each flag category. If you issue multiple names or actions in a single category, XRC processes only the last issued name. For example, the following syntax is incorrect:

```
FLAG   -
      Name(AUTO_READD) Action(ON) -
      Name(SCTRAP) Action(ON)
```

This syntax processes only the Name(SCTRAP) Action(ON) flag. XRC ignores the first name, AUTO_READD. The correct command would issue each name in a separate category:

```
FLAG   -
      Name(AUTO_READD) Action(ON)
FLAG   -
      Name(SCTRAP) Action(ON)
```

"PARMLIB example" on page 154 shows an example of a parmlib command that includes flag names and their action values.

Flag examples

The following is an example of the flag Name(Abend_Lic):

```
FLAG   -
       Name(ABEND_LIC) -
       Action(On)
```

This flag generates a storage dump when the ANTX5106E error message, with return code 638, occurs in the storage control LIC.

The following example shows the flag Name(NRFITF):

```
FLAG   -
       Name(NRFITF) -
       Action(On)
```

This flag generates a storage dump if you receive a 901 type error.

Description and use of patches

Patch parameters temporarily replace existing data with new data to perform corrective action when XRC reaches the specified function. You can determine the current settings for each patch by issuing the XQUERY ENVIRONMENT(PATCH) command. This command displays the current settings for all patches. To determine the default values for the patch names, issue the XQUERY ENVIRONMENT(PATCH) command to an inactive session. This displays the values in the control address space (ANTAS000).

For additional information about the XQUERY command, refer to "XQUERY–Querying a session" on page 90.

For example screens of the XQUERY ENVIRONMENT(PATCH) command, refer to "Examples of XQUERY ENVIRONMENT reports" on page 155.

Patch parameters

The following patch parameters allow you to identify where the patch must be applied.

Name
: Specifies the name of the point in XRC that is to be patched. This is a required parameter.

Offset
: Specifies the offset from the patch point. This is an optional parameter. If this parameter is omitted, the offset defaults to zero.

OldData
: Specifies the currently existing data that is changed by the patch. You can enter up to eight bytes of decimal or character data (for example, OldData(12345678) for decimal or OldData('12345678') for character) and up to four bytes of hexadecimal data (for example, OldData(x'12345678')). The old data is replaced by the new data only if the data in storage matches the old data supplied in this parameter. This is a required parameter.

NewData
: Specifies the new data that replaces the old data. The number of bytes of data

that can be entered is the same as for OldData. However, the length of the old data and the new data must be the same. This is a required parameter.

Note that the patch command will fail if the data in storage does not match either the old data or new data strings. If the data in storage already matches the new data string, the command will be accepted but no changes will be made to the storage.

PATCH syntax considerations

Patch names are issued individually. You must issue only one name in each patch category. If you issue multiple names in a single category, XRC receives only the last issued patch. For example, the following syntax is incorrect:

```
PATCH   -
      Name(DUMPMSG1) OldData('****') NewData('1021') -
      Name(DUMPMSG2) OldData('****') NewData('5001')
```

This syntax passes only the Name (DUMPMSG2) patch. XRC ignores the first patch, Name(DUMPMSG1). The correct syntax issues each name under a separate category:

```
PATCH   -
      Name(DUMPMSG1) OldData('****') NewData('1021')
PATCH   -
      Name(DUMPMSG2) OldData('****') NewData('5001')
```

"PARMLIB example" on page 154 shows an example of a parmlib command that includes patch names and their parameters.

Patch examples

The following example shows a message ID patch:

```
PATCH   -
      Name(DUMPMSG1) -
      OldData('****') -
      NewData('nnnn')
```

This patch causes XRC to dump its storage on a given message ID. For example, if you want a dump when XRC displays message ANTI1021I, code *nnnn* as 1021. After XRC dumps the storage, the message ID number resets to **** to avoid repeated dumps.

Three message ID patches are available:

- DUMPMSG1
- DUMPMSG2
- DUMPMSG3

The following example shows a return code patch:

```
PATCH   -
      Name(DUMPRC1) -
      OldData(X'0000') -
      NewData(X'nnnn')
```

This patch causes XRC to dump its storage on a given return code. For example, if you want a dump when XRC issues return code 607, code X'nnnn' as X'025F' (the hexadecimal equivalent of 607). After XRC dumps the storage, the return code number resets to X'0000' to avoid recursive dumps.

Note: There are six available return code patches. Use the following DUMPRC*n*M patch points if you want to perform a dump when the designated return code appears in a message.

- DUMPRC1
- DUMPRC2
- DUMPRC3
- DUMPRC1M
- DUMPRC2M
- DUMPRC3M

This patch suppresses the creation of a logrec record when an XRC physical session terminates normally.

```
PATCH   -
       Name(S5007P1) -
       OldData(X'00') -
       NewData(X'01')
```

PARMLIB example

The following is an example of member ZOWIE, which was created in parmlib hlq.XCOPY.PARMLIB for use when starting XRC session ZOWIE.

```
/*               ZOWIE                                           */
/*                                                               */
/*  This is an initialization parmlib member for XRC logical session   */
/*  ZOWIE.                                                       */
/*                                                               */
/*  Change Activity:                                             */
/*                                                               */
/*  $L0=OW52938,HDZ11E0,010630,TUCRNC: XRC parmlib support       */
/*                                                               */
VOLINIT -
       MaxNumberInitializations(3) -
       InitializationsPerPrimary(1) -
       InitializationsPerSecondary(1) -
       TracksPerRead(2) -
       TracksPerWrite(2)
SHADOW -
       StorageControlTimeout(00.04.05) -
       MaxBytesTransferred(750001) -
       MaxTracksUpdated(2) -
       MaxTracksFormated(2)
BITMAP -
       DelayTime(00.02.05) -
       ChangedTracks(800)
FLAG   -
       Name(ABEND_LIC) -
       Action(On)
FLAG  -
       Name(SCTRAP) -
       Action(On)
PATCH  -
       Name(DUMPMSG1) -
       OldData('****') -
       NewData('5100')
```

During this session, only three synchronizations and resynchronizations can occur simultaneously. There is a maximum of one simultaneous synchronization per storage control per primary or secondary volume. XRC writes or reads only two tracks at a time. The storage control sessions time out 4 minutes 5 seconds after

XRC stops reading record sets. XRC is restricted to writing 750001 bytes at one time for each channel program. XRC updates or formats only two tracks at one time per write channel program.

The SDM switches the resynchronization bitmaps after the first occurrence of one of the following two conditions:

- 2 minutes 5 seconds have elapsed
- 800 tracks have changed

After the bitmaps have been reset, both the elapsed time and changed tracks counters are also reset to zero.

The flag information indicates that if XRC receives the ANTX5106E, return code 638, error message, it generates a storage dump. Also, XRC requests that the storage control generate a dump of the internal control queue when it receives LIC-related errors.

The patch information indicates that if XRC writes out message ANTS5100E it should generate a storage dump.

Examples of XQUERY ENVIRONMENT reports

To determine the current settings for the parameter values, issue the XQUERY ENVIRONMENT command. The following screens are examples of the output received after entering the command.

XQUERY ENVIRONMENT(FLAG) report

The following example screen shows the output that is received from the following command:

```
XQUERY session_id ENVIRONMENT(FLAG)
```

```
ANTQ8200I XQUERY STARTED FOR SESSION(A) ASNAME(ANTAS001) 882
ANTQ8202I XQUERY ENVIRONMENT_FLAG REPORT - 001
ANTQ8252I NAME          STATUS  NAME          STATUS  NAME          STATUS
ANTQ8203I -------------------------------------------------------------
ANTQ8254I ABEND_LIC        OFF  NRFITF           OFF  SUPRDUMP         OFF
ANTQ8254I AUTO_READD       OFF  SCTRAP           OFF  TIF_ERROR        OFF
ANTQ8254I IGNORE_INTERRUP OFF  SCTRAP2          OFF  XENDDUMP         OFF
ANTQ8203I -------------------------------------------------------------
ANTQ8201I XQUERY ENVIRONMENT_FLAG REPORT COMPLETE FOR SESSION(A)
```

XQUERY ENVIRONMENT(PARM) report

The following example screen shows the output that is received from the following command:

```
XQUERY session_id ENVIRONMENT(PARM)
```

```
ANTL8800I XQUERY SESS1 ENVIRONMENT(PARM)
ANTQ8200I XQUERY STARTED FOR SESSION(SESS1) ASNAME(ANTAS001) 438
ANTQ8202I XQUERY ENVIRONMENT_PARM REPORT - 001
ANTQ8251I NAME                    VALUE  NAME                    VALUE
ANTQ8203I ------------------------------------------------------------
ANTQ8253I zIIPEnable                 NO  MonitorOutput             OFF
ANTQ8253I AllowEnhancedReader        NO  MonitorWakeup           10000
ANTQ8253I BuffersPerStorageCon      576  MHlq                     SYS1
ANTQ8253I ChangedTracks            7500  NoTimeStampCount         5000
ANTQ8253I ClusterMSession      ********  NumberReaderTasks         *,3
ANTQ8253I ClusterName          ********  OfflineDiscovery           NO
ANTQ8253I ConsistencyGroupComb       20  PacingReportThreshol       10
ANTQ8253I DatasetDelay               75  PavByteThreshold       512500
ANTQ8253I DeadSessionDelay           45  PavVolumes                  1
ANTQ8253I DefaultHlq               SYS1  PermanentFixedPages         8
ANTQ8253I DefaultSessionId      DEFAULT  ReaderPacingLimit          33
ANTQ8253I DelayTime            00.30.00  ReaderPacingWindow          3
ANTQ8253I DeviceBlockingThresh       20  ReadDelay                1000
ANTQ8253I DfltWritePacingLvl          0  ReadRecordsPriority       252
ANTQ8253I EnableREFRESHS             NO  ReleaseFixedPages          NO
ANTQ8253I HaltAnyInit                NO  RequireUtility            YES
ANTQ8253I HaltThreshold            1280  ResidualLeftToRead        128
ANTQ8253I Hlq                      SYS1  ScheduleVerify             NO
ANTQ8253I InitializationsPerPr        2  SecondaryDeviceRange   (none)
ANTQ8253I InitializationsPerSe        2  SecondaryVolserPatte   (none)
ANTQ8253I InitializationMethod     FULL  SelectionAlgorithm       LOAD
ANTQ8253I InitializationReadWr      120  ShadowRead                 10
ANTQ8253I IODataAreas               256  ShadowTimeoutPercent       40
ANTQ8253I JournalPriority           251  ShadowWrite                10
ANTQ8253I LowAttention              192  StorageControlTimeou  DEFAULT
ANTQ8253I MaxBytesTransferred    512500  SuppressTimestamp          NO
ANTQ8253I MaxControlTasks           128  SuspendOnLongBusy          NO
ANTQ8253I MaxNumberInitializat        4  SCDumpType           STATESAV
ANTQ8253I MaxTotalReaderTasks        32  TotalBuffers            25000
ANTQ8253I MaxTracksFormatted          0  TracksPerRead               3
ANTQ8253I MaxTracksRead              64  TracksPerWrite              3
ANTQ8253I MaxTracksUpdated            0  UtilityDevice             FIX
ANTQ8253I MinExtenderRead            55  VerifyInterval             24
ANTQ8253I MinLocalRead                0  WorkloadWritePacing  DISABLED
ANTQ8253I MiscHigh                   15  WriteRecordsPriority      253
ANTQ8253I MiscLow                     2  WrtPacingResidualCnt       80
ANTQ8203I ------------------------------------------------------------
ANTQ8201I XQUERY ENVIRONMENT_PARM REPORT COMPLETE FOR SESSION(SESS1)
```

XQUERY ENVIRONMENT(PATCH) report

The following example screen shows the output that is received from the following command:

```
XQUERY session_id ENVIRONMENT(PATCH)
```

```
ANTQ8202I XQUERY ENVIRONMENT_PATCH REPORT - 001
ANTQ8251I NAME                    VALUE  NAME                    VALUE
ANTQ8203I -------------------------------------------------------------
ANTQ8253I DUMPMSG1                 ****  DUMPRC2                  0000
ANTQ8253I DUMPMSG2                 ****  DUMPRC2M                 0000
ANTQ8253I DUMPMSG3                 ****  DUMPRC3                  0000
ANTQ8253I DUMPRC1                  0000  DUMPRC3M                 0000
ANTQ8253I DUMPRC1M                 0000
ANTQ8203I -------------------------------------------------------------
ANTQ8201I XQUERY ENVIRONMENT_PATCH REPORT COMPLETE FOR SESSION(A)
```

Chapter 7. Managing extended remote copy operations

This chapter describes how to manage XRC copy operations.

For additional information about the commands associated with managing XRC copy operations, refer to Chapter 5, “Extended remote copy command descriptions,” on page 69.

In this topic

This topic includes the following sections:

Starting or restarting an XRC session

Use the XSTART command to start an XRC session or to restart a suspended session.

1. You can start an XRC session automatically at system initialization time by putting the JCL for a TSO job, that includes the XSTART command, in a SYS1.PROCLIB member.
2. The operating system then starts the SYS1.PROCLIB member from a SYS1.PARMLIB COMMAND*xx* member.

Note: The storage administrator must issue the XSTART command to start an XRC session if the SYS1.PROCLIB member does not include the XSTART command.

XRC can either be restarted or experience a new start, depending on the steps taken after issuing an XSUSPEND command. XRC will undergo a new start if an XRECOVER command is issued following the XSUSPEND command. If a

XRECOVER command is not issued, XRC can restart. A new start will always occur after you issue an XEND command to end the session; however, a restart is not possible at that point.

When an error occurs and you have specified ERRORLEVEL(SESSION), XRC suspends all volumes in the session but leaves the XRC session active. XRC leaves the secondary volumes in a consistent state within the suspended session. You can correct the error and add the volumes to the active session without having to restart the session. There is minimal impact to the primary application. XRC can rapidly resynchronize the suspended volumes when you add them to the session. This is possible because either hardware bitmaps or software bitmaps maintain the changes to data on the primary volumes.

The following command starts the XRC session with error level set to SESSION. All of the volumes in the session contain associated data. If an error to a volume in duplex status occurs, XRC suspends all volumes in the session. If an error occurs with a coupled, interlocked session, the volumes in all coupled sessions are suspended.

```
XSTART DALLAS SESSIONTYPE(XRC) ERRORLEVEL(SESSION)
```

In the preceding example, the minimum syntax supported is:

```
XSTART DALLAS S(XRC) E(SESSION)
```

The following command starts an XRC session with the error level set to VOLUME. XRC suspends any active volume pair that encounters an error.

```
 XSTART DALLAS SESSIONTYPE(XRC) ERRORLEVEL(VOLUME)
```

Note: When you restart a suspended XRC session with the XSTART command, the system data mover determines which tracks require resynchronization on all of the suspended volumes. The resynchronization process does not occur, however, until you issue an XADDPAIR command for the volume pair.

For additional information about the XSTART command, refer to "XSTART–Starting a session" on page 110.

Workload balancing

Planning for capacity is critical to ensure that there is no performance impact on primary systems because of XRC slowdown. To ensure that the system data mover (SDM) keeps up with update activity generated on primary systems, sufficient capacity in channels, processor resources, processor storage, journal data sets, and secondary bandwidth must be available.

However, even with the best capacity planning, significant deviations from projected workload peaks can still occur. This can occur as a result of unexpected shifts in operation schedules, variations in workloads caused by external conditions, or additional workloads added to an environment, without considering the ramifications to the disaster recovery implementation. In such situations, it is essential to have intelligence built into the SDM solution so it is able to process such peaks with minimal impact to the primary systems, while preserving an XRC environment. These algorithms have been built into the design and are described in greater detail in the sections below.

Application protection

The SDM is designed to have minimal performance impact on the primary system by using a combination of buffering in the software and cache in the storage control. However, environmental or workload conditions can cause either or both of these resources to be consumed, which can result in application performance degradation.

A TIMEOUT value that is specified with the XSET command or with parmlib processing allows you control over application protection versus disaster recovery readiness. The TIMEOUT parameter is often referred to as the *dead system data mover* timer and is used as protection against an extended period of *long busy*. (Long busy is a disk subsystem response to a write I/O request.)

The SDM typically communicates with a disk subsystem 20–30 times per second in a busy environment, and at least once per second in an idle environment. Being unable to communicate for an extended period of time is an unusual situation that can be caused by a line outage, an extended busy lockout condition, or a problem with the SDM.

Each storage control session maintains an independent timer. In an environment where little or no cache updates occur, few cache resources are used by the SDM. In this case, the storage control session may remain active for significantly longer than the TIMEOUT value specified. The maximum time an application can be affected is determined by the TIMEOUT value.

For additional information about the TIMEOUT parameter, refer to “Using the XSET TIMEOUT parameter” on page 183.

Application workload pacing

Certain applications can generate a very large number of updates to a primary volume in a short period of time, temporarily exceeding XRC's capability to offload updates. Such applications include volume defragmentation or full-volume copy or restore operations.

In order to avoid filling cache and reaching Long-Busy conditions, most storage controls have functions that pace the application writes, allowing XRC to offload updates more effectively. The ESS provides such a function, known as device-level blocking (also called device blocking).

Device blocking allows pacing of application writes to occur for brief intervals, and only on the specific volumes that have accumulated a large number of updates in cache. The level of updates that trigger pacing can be tailored using the XRC parmlib. For volumes that contain high-performance response time-sensitive data, such as data base logs, XRC provides the XADDPAIR DONOTBLOCK option that can be used to disable device blocking for an individual volume.

XRC also provides the capability to enable and disable device blocking dynamically for a volume using the XSET DVCBLOCK parameter. This can be used, for example, to permit device blocking for selected volumes during batch processing, and disable it during online processing, all without suspending the mirroring of updates.

In a properly tuned, configured, and balanced XRC environment, device blocking will be infrequent and will have a minimal effect on application performance. If persistent, excessive device blocking is observed, XRC performance data should be analyzed to determine the underlying cause.

Refer to “XADDPAIR–Adding volume pairs or utility volumes” on page 71 and “XSET–Changing session parameters” on page 102 for command syntax, and Appendix C, “ANTRQST and ANTRQSTL macros – call to the system data mover API,” on page 573 for the XRC API invocation of the XADD and XSET requests.

Write pacing

Write Pacing works by injecting a small delay as each XRC record set is created in cache for a given volume. As device residual count increases, so does the magnitude of the pacing, eventually reaching a maximum value at a target residual count. Both this maximum value, and the target residual count at which it is effective, can be specified for each volume through XRC commands. This provides a greater level of flexibility than Device Blocking. Further, the device remains ready to process I/O requests, allowing Application read activity to continue while the device is being paced.

There are fifteen different levels of maximum pacing that you can specify. They are identified by the hexadecimal digits 1 through F, and are specified in XRC commands through the DVCBLOCK keyword values WP1 through WPF. Each volume in the session may be assigned its own value, or you may use a session-level value to apply to a group of volumes. The session-level setting is defined in the XRC PARMLIB value DfltWritePacingLvl, and a special DVCBLOCK value (WP0) is used in XRC commands to assign this attribute to volumes. DfltWritePacingLvl is also used for newly added volumes on write-pacing capable storage subsystems unless overridden on the XADDPAIR command.

WP0 specifies that the session default level will be used, as specified the SHADOW DfltWritePacingLvl PARMLIB value.

WP1-WP7 results in pacing maximums ranging from 0.02 to 2 ms per recordset, and are useful for volumes with high rates of small blocksize writes, such as data base logs and LOGPLUS volumes, where minimal response time impact is essential.

WP8-WPB results in pacing maximums ranging from 5 to 50 ms per recordset, useful for volumes with high mb/sec write rates.

WPC-WPF results in pacing maximums ranging from 100 to 1000 ms per recordset, and should be used in exceptional situations where a high degree of pacing is required.

Target residual count is specified at the session level, through the XRC PARMLIB value WrtPacingResidualCnt. The value is multiplied by 64 to determine the residual count at which the maximum level of pacing will be in effect for write-paced volumes.

Volume blocking and pacing attributes are initially established through the XADDPAIR command, and the presence or absence of the DONOTBLOCK or

DVCBLOCK parameters. Once established, they can be changed for the life of the session with the XSET command using the VOLUME and DVCBLOCK parameters.

Care should be taken in selecting write pacing levels and residual count. Insufficient pacing can result in extended long busy conditions and volume suspension, while excessive pacing can result in unacceptable application throughput and response time.

You can monitor write pacing using the new XQUERY VOLUME_PACE report. Since existing XRC monitors and automation rely on indications of “blocking”, a method has been provided to roughly equate “excessive” pacing to a blocking event. This is done with the XRC PARMLIB PacingReportThreshold value, which specifies the average injected pacing per write I/O, in milliseconds, that must exist for a volume to be considered blocked. This blocking indication is reflected in the usual way at the volume and session level in XQUERY reports, XRC monitor data, and XRC Performance Montior displays and messages.

Workload-based write pacing

Workload-based write pacing is an enhanced form of write pacing that uses MVS workload management (WLM). All the concepts of write pacing, described in “Write pacing” on page 160, apply to workload-based write pacing. However, with workload-based write pacing, write pacing is injected at an application level, allowing more than one level of pacing to be active on a single volume at the same time.

Workload-based write pacing uses the importance value on each write I/O to determine the write pacing level to be used in the calculation for how much write pacing delay to inject, if any. The write pacing threshold is used to determine the write pacing step in the same way. To exploit workload-based write pacing, you must enable it on both the application LPAR and the SDM LPAR. On the application LPAR, WLM must be in use. WLM assigns workloads to service classes with the associated importance levels. For information about setting up WLM, refer to *z/OS MVS Planning: Workload Management* .

Setting up workload-based write pacing

1. Set up WLM on the application LPAR.
2. Specify STORAGESERVERMGT=YES in the IEAOPT*xx* member of parmlib on the application LPAR, so that the writes are correctly classified to XRC.
3. Specify the WorkloadWritePacing parameter in the XRC parmlib values.

Defining WorkloadWritePacing in parmlib

The WorkloadWritePacing parameter has six values corresponding to importance values 1-5 and the discretionary value of 6, meaning no importance. Importance value 1 is the importance value assigned to the SYSTEM service class. Any application that does not have a service class, but is on an LPAR with WLM in use, is assigned to the discretionary importance class. In addition, if XRC is mirroring writes from LPARs that do not support WLM (such as z/Linux or z/VM), or from LPARs that bypass WLM (STORAGESERVERMGT=NO is specified in parnlib member IEAOPT*xx*), then volume-level pacing is used instead. When WorkloadWritePacing is in use on the SDM LPAR and STORAGESERVERMGT=YES is specified in parmlib member IEAOPT*xx* on the application LPAR(s,) Write Pacing is done at an application level and not at the volume level.

DVCBLOCK and DONOTBLOCK

Workload-based write pacing changes the behavior of the DVCBLOCK parameter on XADDPAIR and XSET requests.

When workload-based write pacing is in use, XRC automatically converts DVCBLOCK(ON) to DVCBLOCK(WP*n*), where *n* is the discretionary level, 6.

The value specified with WrtPacingResidualCnt in the XRC parmlib values is used as the threshold, rather than the value for DeviceBlockingThreshold. Refer to "ANTXIN00 parmlib parameters" on page 126 for more information.

DONOTBLOCK applies only when workload-based write pacing is not in use or when the controller is not workload-based write pacing-capable.

For details of DVCBLOCK, DONOTBLOCK and workload-based write pacing, refer to Table 24.

Table 24. DVCBLOCK and DONOTBLOCK Values and Workload-Based Write Pacing

DVCBLOCK Value	Workload-based Write Pacing	Result
DVCBLOCK(ON)	Enabled in parmlib and all requirements are met	DVCBLOCK(WP6), application level pacing
DVCBLOCK(ON)	Enabled in parmlib, WLM disabled or not supported, controller capable	DVCBLOCK(WP6), volume level pacing at discretionary level
DVCBLOCK(ON)	Enabled in parmlib, controller incapable	DVCBLOCK(ON), volume device blocking
DVCBLOCK(ON)	Not enabled in parmlib	DVCBLOCK(ON), volume device blocking
DVCBLOCK(WP*n*)	Enabled in parmlib and all requirements are met	DVCBLOCK(WP*n*), application level pacing
DVCBLOCK(WP*n*)	Enabled in parmlib but requirements not met	DVCBLOCK(WP*n*), volume level pacing at level *n*
DVCBLOCK(WP*n*)	Not enabled in parmlib	DVCBLOCK(WP*n*), volume level pacing at level *n*
DVCBLOCK(OFF) or DONOTBLOCK	Enabled in parmlib and all requirements are met	Application level pacing
DVCBLOCK(OFF) or DONOTBLOCK	Enabled in parmlib, WLM disabled or not supported, controller capable	No blocking or pacing of volume
DVCBLOCK(OFF) or DONOTBLOCK	Enabled in parmlib, controller incapable	No blocking or pacing of volume
DVCBLOCK(OFF) or DONOTBLOCK	Not enabled in parmlib	No blocking or pacing of volume

XQUERY VOLUME PACE

With workload-based write pacing, the XQUERY VOLUME PACE report has several differences:

- It is labeled as REPORT 003, rather than REPORT 002
- Lines are added for the statistics of the levels of the write activity within the volumes.

- The values for RES RATE, WRT RATE, and PACE MS on the detail lines are the total values for the volume.
- When workload-based write pacing is in effect for a storage controller, the threshold count (THD CNT) column separator is changed from `:` to `@`, and the value after this separator is the setting for the write pacing level from an XADDPAIR or XSET command, not necessarily the level at which the volume is being paced.
- An additional detail line is shown for each WLM pacing level (WLM1-WLM6) that identifies what WLM importance levels are getting written to the volume, and the differentiated pacing in effect.

When workload-based write pacing is disabled (with WorkloadWritePacing in parmlib member ANTXIN00), the report reverts to REPORT 002. For more information, refer to “Using XRC reports” on page 184.

To verify whether a controller is capable of workload-based write pacing, use the XQUERY STORAGECONTROL XFEATURES command instead of the XQUERY STORAGECONTROL FEATURES command. For more information, refer to “Using XRC reports” on page 184.

Subsystem long-busy state

If the SDM read process stalls, or if the application update rate continually exceeds the capability of the SDM to offload application updates, the amount of cache consumed in the subsystem will continue to grow.

If the SDM continues to read data, except at a relatively slow rate, the *dead system data mover* timer, discussed in “Application protection” on page 159, will not be invoked because the SDM is still running, but not keeping pace with the cache consumption rate.

Over time, the residuals will grow for all primary volumes and a long busy condition may be reached. This condition can result from excessive cache consumption (on an IBM ESS, copy service functions are limited to 60% of the cache resource) or an excessive number of records in cache for a particular scsession (on the IBM ESS, the limit is 63K records).

It is possible for the primary disk subsystem to remain in a long-busy condition for 80% of the TIMEOUT value for a given session. When this occurs, the SDM suspends the volumes in the session. If this action does not correct the condition, the session ends for non-ESS subsystems and is suspended for ESS subsystems.

Synchronization and resynchronization processing

Synchronization and resynchronization functions are designed to occur at any time. This function has the intent of restoring recovery capability in as short a time as possible.

The synchronization process stops if the residual count builds up on the primary subsystem. It is possible for the total number of residuals for a single storage control session to exceed a specified value. When this occurs, the SDM suspends all synchronization activity for volumes in that SSID until the residual count drops below the threshold. This prevents any significant delays caused by the synchronization process. It also ensures that you do not have to wait for a period of low activity to perform this process. The HaltThreshold parmlib parameter controls this residual count level.

Both planned and unplanned outage support use volume level bitmaps to provide a partial resynchronization capability where only changed tracks are copied. While a volume pair is in duplex status, the bitmaps are continually updated to maintain a record of tracks that have changed data. These bitmaps must be refreshed periodically to minimize the amount of data that must be copied during the resynchronization process.

Synchronization

The options available when you initially add a primary volume to an XRC session include FULLCOPY, QUICKCOPY, and NOCOPY. The choice to use the FULLCOPY or QUICKCOPY options for the synchronization process depends on a number of factors, which are described in "Adding a volume with the FULLCOPY or QUICKCOPY option" on page 172.

The SYNCH and SCSYNCH parameters control the number of synchronization tasks that are initiated simultaneously and the tasks that are allocated over the available SSIDs. These options are global for an SDM. In a heterogeneous environment, it may be necessary to set these options based on one particular configuration.

The main consideration for the synchronization options is the available bandwidth between the primary subsystems and the SDM. Configure the environment to fully use this bandwidth so long as this does not cause performance problems due to the increased read activity on the primary subsystems. Although the SDM halts the synchronization process if under stress, concurrent update activity can still occur. The specified SYNCH and SCSYNCH values also depend on the amount of virtual storage available in the LPAR in which the SDM is running. For each synchronization task that is running, the SDM requires 300 kilobytes of real storage.

Set the SCSYNCH option to the highest value calculated for different disk subsystems (or slightly less than this value) to ensure the fastest overall resynchronization. This will be achieved if the subsystems finish a full synchronization at approximately equal times, or if the subsystems with longer synchronization times are able to use the bandwidth of those that have finished earlier.

Because the throughput of a single task depends on a particular environment, it is best to determine optimum synchronization parameters by experimentation. You can use a method similar to the following:

1. Synchronize a single volume and determine the percentage of available bandwidth, Enterprise System Connection (ESCON), FICON, channel extender, and Wide Area Network (WAN) that is used. Your test should contain representative data because unrealistic figures can result if compression is used on the network.
2. Determine how many volumes, at this rate, would use all available bandwidth and determine the synchronization parameters.
3. Run a series of tests using these options with this value to determine the optimum values.

Halt of synchronization while under stress: If the primary subsystem residual count rises above a certain threshold, the SDM halts the synchronization activity until the residual count is below the threshold.

It is not necessary to wait for periods of low data activity to perform synchronization. However, synchronizing through a peak workload will take longer than during a period of relative inactivity.

Resynchronization

Both planned and unplanned outage support use volume level bitmaps to provide a partial resynchronization capability where only changed tracks are copied. While a volume pair is in duplex status, the bitmaps are continually updated to maintain a record of tracks that have changed data. These bitmaps must be refreshed periodically to minimize the amount of data that must be copied during the resynchronization process.

Note: For resynchronization operations to complete faster, ensure that XRC sessions remain uncoupled during the resynchronization process.

This refresh process is controlled by the RTRACKS (the number of updated tracks) and RFREQUENCY (time) options of the XSET command. You should not refresh the bitmaps too often because it slows SDM performance for large numbers of volumes. When a subsequent XADDPAIR command is issued for a suspended volume, the SDM will read the hardware bitmaps and only copy the changed tracks.

Using XRC utility devices

This section discusses utility devices that XRC uses to read from a primary storage control. When an SDM reader task reads changed data from a primary subsystem, it must use a device address available on that subsystem. There is an SDM reader subtask for each storage control session and, at a minimum, there is one storage control session associated with each SSID. The SDM requires a device address in each storage control session, which is called a utility device or reader.

Any device address that is associated with the storage control session can be used as the utility device. If the device is used for any significant amount of primary system I/O activity, application activity will block the SDM from transferring data. This situation can result in the SDM experiencing performance delays.

Conversely, when the microcode designates a particular device as its utility device, it has a special status that prevents if from being used by the primary systems. This may also have an impact on primary systems.

You can define utility devices by using the XADDPAIR command for a single cylinder volume, with the XRCUTL specified for the secondary volume. If custom volume sizes are supported by the subsystem, then single cylinder volumes can be created for use as utility devices. You can define multiple utility volumes for a storage control session. (Eighty percent or more of customers use fixed utility devices.)

Addressing the utility device selection

The XSET command allows you to specify the method by which XRC selects utility devices for use in reading data from a storage control. You can choose that a specific device be used. This is known as a fixed device. Or you can allow the storage control to determine which utility device to use. This is known as a floating device.

The utility device method is controlled through the XSET command, and the utility device to be used is specified using the XADDPAIR command with the secondary volser of XRCUTL. The UTILITY parameter of the XSET command, when used in conjunction with the XADDPAIR command, allows you to:

- Set up a storage control session to use a particular device as a fixed utility device.
- Change to a different fixed utility device for a storage control session.
- Disable the fixed utility support for a particular storage control session and return to the default floating utility device support.

Note: Be aware that issuing the XSET command with the UTILITY parameter specified does not activate or deactivate fixed utility device support. This support is enabled when you issue an XADDPAIR command with a secondary volser of XRCUTL.

For additional information about utility devices, refer to "Fixed utility devices" and "Floating utility devices" on page 167.

Fixed utility devices

For fixed utility devices, XRC always reads data from the storage control using this device, thus eliminating application contention for devices. The following recommendations apply to these devices:

- Select dedicated fixed utility devices that are not used for any other purpose.
- Ensure that the device is not reserved by other applications. This is not a concern on 2105 subsystems; it only applies for non-2105 (3990 subsystems, for example).
- Ensure that the first device added to a session for a particular storage control be a fixed utility device.
- Define the fixed utility device as a single cylinder device if a storage control is an IBM 2105 ESS. Specify a single fixed utility device per storage control session.

Under certain conditions, you may want to remove potential interference between the primary system and the SDM. You can do this by defining a dedicated (fixed) utility device address per storage control session. Fixed utility devices are preferred if:

- You have a few volumes available to select from
- You are running in a channel-extender environment
- You want to isolate the SDM I/O to a single primary volume for ease of monitoring, tracing, or debugging
- You want to isolate the SDM I/O so it does not interfere with any primary system I/O

In a channel-extender configuration, it is highly recommended that you define fixed rather than floating utility devices for the following reasons:

- With a channel-extended configuration, telecommunication links tend to be the most vulnerable. If they fail, it is essential that you do not leave a production volume in the status of an XRC utility device. If an application attempts to reserve a primary volume that is currently being used as a utility device, message `IEA482I` is written to the system log (SYSLOG). The application program will not be able to proceed until the storage control session is suspended.

- The channel extender definition and error recovery are at a device level. Error recovery works better if a single device is used by the SDM to read the updates for each storage control session because of the method that is used to communicate errors to IOS.
- You can vary all nonutility devices offline to the SDM so that your Resource Management Facility (RMF™) device activity report contains only the utility device status.
- You can define utility devices on additional paths to improve resilience and performance.

To move from floating to fixed utility devices, it is not necessary to end or suspend an XRC session. You can issue the XSET UTILITY(FIX) command, and then issue the XADDPAIR command for the designated XRCUTL device (even if the device is already in the session).

Table 25 describes device utility support when the XSET command is specified with UTILITY(FIX) and the XADDPAIR command with a secondary volser of XRCUTL is specified.

Table 25. Specifying UTILITY(FIX) and the XADDPAIR command with a secondary volser specifies as XRCUTL

If . . .	Then . . .
The primary volser is *not* in the session.	It is added and it becomes the fixed utility device.
The primary volser is already in the session, its secondary volser is XRCUTL, and there is no outstanding XDELPAIR command issued against it.	The primary volser becomes the fixed utility device.
The primary volser is already in the session, its secondary volser is XRCUTL and there is an outstanding XDELPAIR command issued against it.	The XADDPAIR command is rejected, with return code 4027.
The primary volser is already in the session and its secondary volser is *not* XRCUTL.	The XADDPAIR command is rejected, with return code 489.

Floating utility devices

For floating utility devices, XRC allows the storage control to indicate which device to use when reading data from the storage control.

The floating utility device, a cooperative function between the subsystem and the SDM, was created so that a dedicated real device would not be required for a utility device. Floating utility devices should be used if you have insufficient devices to dedicate for the exclusive use of the SDM. Contact the subsystem vendor to determine which options are supported for other subsystems.

Note: Floating utility devices are not supported for enhanced multiple reader sessions.

Primary subsystems that support floating utility devices continuously evaluate all volumes in the session based on their activity. At the request of the SDM, the best utility device candidate is selected and that device address is used to read updates from the storage control session.

With floating utility device support, any volume that is added is a candidate to be used as a utility device. If you use floating utility devices, it is critical that a sufficient number of candidate volumes be available to float among. It is recommended that you define no fewer than four volumes per reader, per storage control session.

If a volume is reserved by an application program, it is ineligible to be used as a utility volume until it is released. In a typical environment, if fewer than four volumes are used, there is a high probability of deadlock between the SDM and the application reserve activity. If there are sufficient volumes in a session, the SDM should always have a suitable utility volume available for use.

If an application attempts to reserve a primary volume that is currently being used as a utility device, and this device cannot be released by the microcode because there are no other utility device candidates, message `IEA482I WAITING FOR CONTROL UNITS` is written to the SYSLOG. The application program will not be able to proceed until another utility volume is selected by the disk subsystem microcode.

To move from fixed to floating utility devices, it is not necessary to end or suspend the XRC session. You can issue the XSET UTILITY(FLOAT) command, and then reissue an XADDPAIR command for a single device in all storage control sessions.

Table 26 describes utility support when the XSET command is specified with UTILITY(FLOAT) and the XADDPAIR command with a secondary volser of XRCUTL is specified.

Table 26. Specifying UTILITY(FLOAT) and the XADDPAIR command with a secondary volser of XRCUTL

If . . .	Then . . .
The primary volser is *not* in the session.	It is added to the session. The fixed utility mode is deactivated if it is active and support returns to floating utility mode.
The primary volser is already in the session and its secondary volser is XRCUTL.	The fixed utility mode is deactivated if it is active and support returns to floating utility mode.
The primary volser is already in the session and its secondary volser is *not* XRCUTL.	The XADDPAIR command is rejected, with return code 489.

Re-enabling floating utility devices when XRC incremental resynchronization is disabled:

About this task

Incremental resynchronization requires fixed utility devices. When you disable incremental resynchronization, you cannot transition enhanced readers from fixed to floating. However, if you are using single-reader readers, you can convert existing fixed utility devices to floating utility devices as follows:

Procedure

1. Issue the XSUSPEND command with the TIMEOUT keyword.
2. Issue the XSTART command with MODE(NORMAL).
3. Issue the XSET command with UTIL(FLOAT).

4. Issue the XADD command for each utility device already in the session to disable the SWAP devices.
5. Issue the XADD SUSPENDED command to make data devices active.
6. Modify PARMLIB to include UITLITYDEVICE(FLOAT). This ensures that future restarts maintain the utility devices as floating devices.

More about utility devices

Table 27 describes utility support when the UTILITY parameter is specified with either FIX or FLOAT and the XADDPAIR command with a secondary volser is not XRCUTL:

Table 27. Specifying UTILITY(FIX or FLOAT) and the XADDPAIR command with a secondary volser that is not XRCUTL

If . . .	Then . . .
The primary volser is *not* in the session.	The primary volser is added to the session.
The primary volser is already in the session and its secondary volser is XRCUTL, but it is not a fixed utility device.	The XADDPAIR command processes normal volume initialization.
The primary volser is already in the session, its secondary volser is XRCUTL and it is a fixed utility device.	The XADDPAIR command is rejected, with return code 4082.
The primary volser is already in the session and its secondary volser is not XRCUTL.	The XADDPAIR command is rejected with either: • Return code 490, if the new secondary volser matches an existing secondary volser. • Return code 489, if the new secondary volser is different that an already existing secondary volser.

If the following situations occur, a storage control session for a utility volume is converted to an Enhanced Auxiliary session:

- AllowEnhancedReader(YES) is specified.
- The storage control session is currently in SINGLE mode.
- The SCSESSION value is different from the value originally specified for the volume.
- The new SCSESSION value corresponds to an Enhanced Primary session.

If the following situations occur, a storage control session for a utility volume is converted to an Enhanced Primary session:

- AllowEnhancedReader(YES) is specified.
- The storage control session is in SINGLE mode.
- The SCSESSION value is the same as the value originally specified.

Storage control sessions suspend whenever all volume pairs in the XRC session are suspended. As a result, utility volumes (XRCUTL) suspend as well. In addition, when the last data volume is suspended in a storage control session, this causes the utilities associated with that storage control session to suspend as well. Further, anytime a data volume is suspended, any storage control session that has no data volumes associated with it will also be suspended.

Using parallel access volumes for utility devices

When you are using PAV for utility devices through the use of the NumberReaderTasks parameter with a nonzero value, the alias devices must be continuously available for XRC use. Dynamic alias management does not provide the required continuous alias availability and should not be used for XRC utility volumes. Use either static alias assignment or HyperPAV.

Adding an XRC volume pair

The XADDPAIR TSO command specifies both the primary (source) and secondary (target) volumes of a volume pair that you want to add to an XRC session. You can add up to 50 volume pairs with each XADDPAIR command.

Note:

1. When you add a suspended volume pair back to the session, the data on the recovery (target) volume is incomplete (not ready for recovery) until the volume pair reaches duplex state. Choose the resynchronization time carefully to allow the volume pairs to quickly reach duplex state.
2. After XRC processes an XADDPAIR command, it delays the start of volume initialization slightly to allow you time to issue additional XADDPAIR commands.
3. A high-activity level on the primary volume's storage control can delay the initial volume copy process. XRC does not start, or continue with, an initial volume copy or resynchronization process on a storage control that exceeds the update threshold (HaltThreshold parmlib parameter). If the volume pair's copy-completed percentage is not changing as expected, look at the STA field of the XQUERY VOLUME report. If the status is "pnd" or "cpy" the copying has halted. If the status is "PND" or "CPY" the copying has not halted. When the activity level declines, the volume copy continues.
4. When adding a volume pair using the LOGPLUS parameter with the XADDPAIR command, be aware that any volume previously activated by System Logger will be suspended by XRC if System Logger is not currently active. If XADDPAIR results in the suspension of a specified volume, issue an XQUERY and check the volume status. L+ indicates that the volume was activated by System Logger. L- indicates no System Logger connection for that volume.

   This situation may also occur when volume pairs are re-added into a session using the SUSPENDED parameter of the XADDPAIR command if one or more of the volume pairs was initially added with the LOGPLUS parameter.

   If you find that XRC has suspended a L+ volume due to System Logger inactivity, you can use the ICKDSF REFORMAT command to clear the L+ flag in the volume label. More information can be found in *Device Support Facilities (ICKDSF) User's Guide and Reference* .
5. When adding LOGPLUS volumes, you may want to minimize impact to write response time. Refer to "Adding a volume with the LOGPLUS option" on page 174.
6. If a volume pair has been removed from the session with XDELPAIR, it cannot be re-added until residual updates for the volume have been drained from the primary storage subsystem cache. This process can take several seconds to several minutes to complete, depending on the current XRC session delay.

For additional information about the XADDPAIR command and to review the XADDPAIR command syntax, refer to "XADDPAIR–Adding volume pairs or utility volumes" on page 71.

Initializing volume pairs

The XRC volume pair initialization function selects the next volume to synchronize or resynchronize, and prioritizes the volumes from the command's list of volumes. The first priority is to resynchronize existing volume pairs, then to add new volume pairs to the XRC session. XRC selects and adds new pairs as needed according to the PRIORITY subparameter (FIFO or LOAD), which you can specify with the XSET command. See Table 28 for more information.

Table 28. PRIORITY specification with the XADDPAIR command

If you specify . . .	Then . . .	Notes . . .
PRIORITY(FIFO)	XRC processes the volumes that were listed by an XADDPAIR command in the order that they are listed.	1. First, XRC processes all volumes that are eligible for resynchronization. 2. Next, XRC processes all volumes that are eligible for synchronization. XRC processes the volumes within the limits that you specified in the SYNCH and SCSYNCH parameters of the XSET command.
PRIORITY(LOAD)	XRC processes the volumes that were listed by an XADDPAIR command based on the primary and secondary storage control load in this order:	1. First, XRC processes all volumes that are eligible for resynchronization. XRC can defer the initialization if the associated storage control has too much activity at the time that the volume is eligible. The volume continues to be eligible for processing when the activity level has decreased. 2. Next, XRC processes all volumes that are eligible for synchronization. Again, XRC can defer the initialization if the primary storage control activity is too high. XRC performs both resynchronization and synchronization processing within the limits that the SYNCH and SCSYNCH parameters of the XEND command specify.

Adding volumes in a volume group

With the ERRORLEVEL keyword of the XADDPAIR command, you can assign a common group name to volumes that are related. XRC suspends all of the volumes in a common group when an error occurs to any one of the volumes that belong to that group.

For additional information about adding related volumes as groups, refer to "XADDPAIR command syntax" on page 71.

For additional information about the ERRORLEVEL parameter, refer to Chapter 11, "Recovering from error conditions using extended remote copy," on page 255.

Adding a volume with the FULLCOPY or QUICKCOPY option

Unless you specify NOCOPY, XRC makes an initial copy of the primary volume. Use Table 29 to determine which copy option to choose.

Table 29. Copy option determination

If you use . . .	Then . . .	Notes . . .
The default FULLCOPY option	XRC copies all tracks on the primary volume to the secondary volume. As each track is copied, the primary subsystem adds the track to the storage control session and starts monitoring that track for updates to ensure they are captured during the copy process. The SDM does this by reading a number of tracks in a single channel command word (CCW) chain and, in the same chain, adding those tracks to the storage control session. If you are using the FULLCOPY option for tracks that have never been written or that contain an end-of-track indicator, data will not be transmitted (it is an empty track). For tracks that have been written to or are in the volume table of contents (VTOC), data will be transmitted.	You may change the default by using the XSET COPY option. See "XSET command syntax" on page 103.
The QUICKCOPY option	XRC copies *only* allocated tracks on the primary volume to the secondary volume during this initialization of the secondary volume. Initialization of the secondary volume is performed while copying primary volume updates to the secondary volume.	During this processing, XRC does not copy the volume serial number from the primary volume to the secondary volume. Therefore, during normal copy operations, a display of the secondary volume serial number shows its original serial number. During XRECOVER processing, XRC sets the secondary volume serial number equal to the serial number of the primary volume; this is not performed for XADVANCE processing.

Note:

1. The choice to use the FULLCOPY or QUICKCOPY options for the synchronization process depends on a number of factors. If you are using channel-extension technology, you should use the FULLCOPY option as there is risk that a reserve could be left outstanding on the primary device if the network fails during the copying process.
2. XRC ensures that it has successfully copied all tracks (with FULLCOPY) or all allocated tracks (with QUICKCOPY) on the primary volume, and all updates that are made to the volume during initialization, to the secondary volume. As a result, when volume initialization is complete, the secondary volume is in a known, consistent state.
3. If you have a restricted amount of bandwidth between the primary and secondary devices, then specifying the QUICKCOPY option may be extremely beneficial. Performing a large number of full synchronizations of primary and secondary devices increases the benefit of using the QUICKCOPY option.
4. As part of the resynchronization function, XRC automatically copies all changed and any not previously copied tracks whenever suspended volume pairs are added again to a session. For this resynchronization function, XRC ignores any copy option specified.

Use the QUICKCOPY option with caution

As a part of QUICKCOPY initial processing, XRC must issue a reserve for the primary volume. After the tracks that contain the allocated space for the volume have been identified, XRC issues a release during the initial phase synchronization phase. If access to the primary volume is through a channel extender and the connection to the primary volume is lost while XRC has the reserve, XRC will not be able to release the volume. Applications then will not be able to access the primary volume.

Adding a volume with the NOCOPY option

The NOCOPY option bypasses the initial primary to secondary volume copy and eliminates the initial data transfer between volumes. This lets you quickly initialize a large number of XRC-managed volumes.

Use the NOCOPY option with caution

Because of potential data integrity exposures, use the NOCOPY option of the XADDPAIR command only in TSO command mode, not in automated procedures. Specify NOCOPY only when installation procedures can guarantee that the primary and secondary volumes are equivalent at the time that you issue the command.

The NOCOPY option specifies that the primary and secondary volumes are identical and that it is not necessary to copy the primary volume to the recovery system. Be careful when you specify the NOCOPY option when any of the following are true:

- You issue the XADDPAIR command as part of automatically initiated XRC subsystem activity, because the NOCOPY option can introduce potential data integrity exposures during a system failure.

  Example: An application may have completed a successful write operation to a primary volume by placing the data in the storage control's nonvolatile storage. Before the data mover can copy the data to the recovery system, however, the SDM system fails. If you specify NOCOPY when you restart the SDM system, XRC does *not* copy the primary volume to the secondary volume. XRC will not synchronize them even though the two volumes may no longer contain exactly the same data. There is a data integrity exposure here because latent updates exist, the system data mover has not received the updates, and there is no indication of missing data.
- The XADDPAIR command follows a DFSMSdss full-volume restore that included the COPYVOLID parameter. In this case, you must manually set the secondary volume serial number to a unique ID, and vary it online.

Note: The XADDPAIR TSO command with the NOCOPY option requires a few seconds to complete across the sysplex. If application hosts initiate disk write updates to the volumes before the volumes are under XRC control, those updates will not be timestamped. (Only write I/O operations to volumes that are part of an XRC session are timestamped.) This temporary inconsistency has passed when an XQUERY command returns the information that there are no volumes in seqcheck status.

The XADDPAIR command with NOCOPY is equivalent to the dual copy function with the NOCOPY option. **You must use it with care**. When you specify NOCOPY, both the primary and secondary volumes must contain exactly the same data at the point in time when the pair is established. To ensure that the volume contents match, perform the following steps:

1. Stop all application activity to the primary volumes.
2. Use DFSMSdss to dump the full primary volumes. Specify the ALLDATA and ALLEXCP options to dump both allocated used and allocated unused space on the volumes. (Some applications refer to the allocated unused space of a data set.)
3. Restore the full secondary volumes.
4. Ensure that no applications have made updates to the primary volumes during the volume dump and restore process.
5. Issue the XADDPAIR command with the NOCOPY option.
6. Once the XRC pair is established, restart the applications to the primary volumes. If this is part of an automated procedure, you can check for the ANTA8101I or ANTA8004I message to detect when the pair is under XRC control.

Adding a volume with the SUSPENDED option

Use the XADDPAIR command SUSPENDED keyword to restore the volume configuration to the state that existed prior to a suspension. The keyword processes all pairs that are suspended at the time the XADDPAIR command is issued. XRC then places each primary volume in an active state so that it is eligible to start resynchronization, and so that each secondary volume can receive updates.

Example: Say that there are 1000 volumes in the XRC session and 200 are in suspended state and 800 are in duplex state. An XADDPAIR SUSPENDED command would start the volume initialization process for the 200 suspended volumes. XRC initializes the volumes subject to the SYNCH and SCSYNCH values that are currently in effect.

Adding a volume with the DONOTBLOCK option

Specify the DONOTBLOCK option to exclude a volume from device-level blocking. The SDM uses the device-level blocking function when the primary storage control supports this function. An application might generate significant updates to a primary volume in a single-channel program chain, such as a defragmentation or full-volume restore operation. The device-level blocking algorithm paces the operation to allow the SDM to offload updates efficiently. Consider using the DONOTBLOCK option only for high-performance, sensitive volumes that generate very short updates, such as IMS™ write any data set (WADS) volumes.

Workload-based write pacing affects the behavior of DONOTBLOCK. For more information, refer to "Workload-based write pacing" on page 161.

Adding a volume with the LOGPLUS option

LOGPLUS specifies that the primary volume of a volume pair is to be explicitly written to by the z/OS System Logger. When adding a volume pair using the LOGPLUS parameter with the XADDPAIR command, be aware that any volume previously activated by System Logger will be suspended by XRC if System Logger is not currently active. If XADDPAIR results in the suspension of a specified volume, issue an XQUERY and check the volume status. L+ indicates that the volume was activated by System Logger. L- indicates no System Logger connection for that volume.

This situation may also occur when volume pairs are re-added into a session using the SUSPENDED parameter of the XADDPAIR command if one or more of the volume pairs was initially added with the LOGPLUS parameter.

If you find that XRC has suspended a L+ volume due to System Logger inactivity, you can use the ICKDSF REFORMAT command to clear the L+ flag in the volume label. More information can be found in *Device Support Facilities (ICKDSF) User's Guide and Reference*.

Only a single volume pair may be specified on an XADDPAIR command or XADD API request if LOGPLUS is also specified. You may specify a single utility volume with this pair. The utility volume must be specified after the volume pair on the VOLUME parameter.

When adding LOGPLUS volumes, you should take care to specify one of the following in the XADDPAIR command to minimize impact to write response time:

- DONOTBLOCK or, if the volume resides in a storage subsystem supporting write pacing
- A low write pacing value of 1 - 7 specified with the DVCBLOCK(WP*n*) parameter.

Note: DVCBLOCK and DONOTBLOCK are affected by workload-based write pacing. For more information, refer to "Workload-based write pacing" on page 161.

A LOGPLUS storage control session cannot have an Enhanced Auxiliary session. Only an Enhanced Primary session with no auxiliary sessions, or a Single mode session is supported for LOGPLUS volumes.

For more information on System Logger, refer to the topic titled Plan DRXRC-type Staging Data Sets for Coupling Facility Log Streams in *z/OS MVS Setting Up a Sysplex*.

Understanding the XADDPAIR command

The following are examples of XADDPAIR commands:

- To add utility volumes and regular volume pairs to two storage controls, use the following commands:

```
XADDPAIR DALLAS VOLUME(PRIMA1 XRCUTL,PRIMB1 XRCUTL)
    XADDPAIR DALLAS VOLUME(PRIMA2 SEC001,PRIMB2 SEC002)
```

- To add utility volumes and volume pairs to two storage controls, use the following command:

```
XADDPAIR DALLAS VOLUME(PRIMA1 XRCUTL,PRIMB1 XRCUTL,PRIMB2 SEC001,
           PRIMB2 SEC002)
```

- To copy primary volumes PRIM01 and PRIM02 to secondary volumes SECO01 and SECO02 to an XRC session called Dallas, issue the following command. A blank or comma is used to separate each primary and secondary volume serial number within each pair; a comma is used to separate the pairs.

```
XADDPAIR DALLAS VOLUME(PRIM01 SECO01,PRIM02 SECO02)
```

- To add a volume pair to an XRC session called Dallas, route XRC messages to user ID E299000, and specify that XRC only copy updates to the primary volume, issue the following command. XRC does not copy the initial volume image

```
XADDPAIR DALLAS VOLUME(PRIM01 SEC001,PRIM02 SEC002)
              MSGROUTEID(E299000) NOCOPY
```

Note: XRC ignores the NOCOPY keyword, however, if you issue this XADDPAIR command to resynchronize the volume pair.

- To add volume pairs to an XRC session called Dallas so that both pairs belong to a group that is called DB2GROUP, issue the following command:

```
XADDPAIR DALLAS VOLUME(PRIM01 SEC001,PRIM02 SEC002)
             ERRORLEVEL(DB2GROUP)
```

- To add volume pairs with an SCSESSION keyword, use the following command. In this example, the command adds two volume pairs to the storage control session called GA. The XADDPAIR SCSESSION keyword provides a way to manage volumes for a single storage control. If there are multiple storage control sessions within that storage control, then all of the sessions are affected.

```
XADDPAIR DALLAS VOLUME(PRIM01 SEC001,PRIM02 SEC002)
             SCSESSION(GA)
```

Guideline: For the best SDM performance, assign heavy use volumes to separate storage control sessions.

Example: use JCL to start an XRC session

You can run the following JCL example as a batch job, or process it from a SYS1.PROCLIB member to automate the procedures that are shown here. This example shows how to use JCL to start an XRC session, add a single volume pair to the session, and route message output to a specific user ID.

```
//IKJEFT01  JOB,
     //         MSGLEVEL=(1,1),MSGCLASS=A,NOTIFY=user ID
     //STARTXRC  EXEC PGM=IKJEFT01
     //STEPLIB DD DSN=authorized.cmdlib,DISP=SHR
     //SYSTSPRT DD SYSOUT=*
     //SYSTSIN  DD *
       PROFILE MSGID
       XSTART DALLAS SESSIONTYPE(XRC) ERRORLEVEL(VOLUME) MSGROUTEID(user ID)
       XADDPAIR DALLAS VOLUME(PRIM01,SEC001) MSGROUTEID(user ID)
     /*
```

Example: use a CLIST to start an XRC session

You can process the following TSO CLIST in a standard TSO environment. This example shows how to use a CLIST to start an XRC session and add two volume pairs to the session.

```
PROFILE MSGID
     XSTART DALLAS SESSIONTYPE(XRC) ERRORLEVEL(VOLUME)
     XADDPAIR DALLAS VOLUME(PRIM01,SEC001,PRIM02,SEC02)
```

Using the ATTIME, DRAIN, and IMMEDIATE keywords

The ATTIME, DRAIN, and IMMEDIATE keywords let you control the consistency group time when you delete, end, or suspend volume pairs. Consider the following guidelines when using these functions.

- Use the ATTIME keyword in environments where applications continue to update the primary volumes even after the time specified on the ATTIME keyword has passed. XRC will not apply primary updates to the secondary volumes beyond the time you have specified on the ATTIME keyword.
- XRC processes timestamps with respect to the timestamp that is generated by updates on the primary system. Depending on the SDM system load, the XRC session consistency group time for the primary system may lag behind the SDM system time. As a result, ATTIME events can sometimes lag behind the SDM system time.
- The ATTIME and DRAIN keywords do not apply to volumes that are in the suspended state. You can only delete these volumes with an XDELPAIR IMMEDIATE command or an XEND command.
- Specify the VOLUME(ALL) and IMMEDIATE keywords if you want to process all volumes in the session, regardless of their volume states. The exception is that XSUSPEND ALL IMMEDIATE will *not* include utility volumes or volumes that are already suspended.
- It is possible that all volume pairs can become deleted or suspended, either due to an error or by command, either before the specified ATTIME occurs or before the system data mover can DRAIN the stored updates from the primary storage control. If this happens, XRC cancels the command that contains the affected ATTIME or DRAIN parameter, and clears any timestamp that is related to that command.
- Under certain circumstances, the SDM must approximate the time when the specified ATTIME occurs on the primary system. This can happen, for example, if you specify an ATTIME value with a command, but the application activity to the primary volumes has already been quiesced. The current timestamp of the primary system is therefore unavailable to the data mover, so the SDM must approximate the time. XRC conservatively approximates the time to ensure that enough time has passed on the primary system. XRC does not apply data to the secondary volumes beyond the specified ATTIME. The ATTIME event may therefore happen later than expected. XRC guarantees that the data on all secondary volumes is consistent as of the reported ATTIME.
- When you issue XSUSPEND or XEND commands to a master session, the corresponding action is processed on all XRC sessions that are coupled to the master session. The IMMEDIATE parameter is processed as ATTIME equal to a time that permits all coupled sessions to be ended at the same consistency time. The ATTIME parameter is processed as global ATTIME if no component session consistency time has passed the requested timestamp. If any session has passed the time, the command is rejected. The DRAIN parameter is processed as ATTIME by using the latest component drain time.

Deleting an XRC volume pair

Use the XDELPAIR command to delete volumes or utility devices from an XRC session.

The XDELPAIR command returns a timestamp that identifies the application time that the volume was last consistent with the XRC session. After you issue the XDELPAIR command, XRC prompts you (unless you specified NOVERIFY) to confirm the request to delete the volumes. You can start the XDELPAIR command through a CLIST in either a foreground or a batch environment. If you do, you should either specify NOVERIFY or place your response to the prompt in the data stack. See the CLIST examples that follow.

If a volume pair has been removed from the session with XDELPAIR, it cannot be re-added until residual updates for the volume have been drained from the primary storage subsystem cache. This process can take several seconds to several minutes to complete, depending on the current XRC session delay.

Examples: The following are examples of commands to delete one or more volume pairs:

- The following command deletes a volume pair from the XRC session:

```
XDELPAIR DALLAS VOLUME(PRIM01)
```

- The following command deletes three volume pairs from the XRC session and routes messages to user ID E299000:

```
XDELPAIR DALLAS VOLUME(PRIM01)
```

```
XDELPAIR DALLAS VOLUME(PRIM01 PRIM02 PRIM03) MSGROUTEID(E299000)
```

- You can process the following TSO CLIST in a standard TSO environment for commands that require a response. The following example shows how to use a CLIST to delete two volume pairs:

```
PROFILE MSGID
     CONTROL PROMPT
     XDELPAIR DALLAS VOLUME(PRIM01,PRIM02)
     DATA PROMPT
     YES
     ENDDATA
```

- The following REXX EXEC example adds a new volume pair to the XRC session and deletes another pair from the session:

```
/* REXX */
   'PROFILE MSGID'
   'PROFILE PROMPT'
   SETPROMPT=PROMPT('ON')
   'XADD   A V(XRCF40,SECF51)'
   PUSH 'YES'
   'XDEL   A V(XRCF26)'
   EXIT
```

- You can submit the following batch job to the system:

```
//XRCJOB   JOB   MSGLEVEL=(1,1),MSGCLASS=H,REGION=2048K,CLASS=A
    //XSTART   EXEC  PGM=IKJEFT01
    //STEPLIB  DD    DSN=IBMUSER.CLIST.CLIST,DISP=SHR
    //SYSTSPRT DD    SYSOUT=*
    //SYSTSIN  DD    *
      PROFILE MSGID
      EXEC 'TSOUSER.CLIST.CLIST(JOB)' C
    /*
```

- The above JCL starts the following CLIST. The CLIST is in the data set TSOUSER.CLIST.CLIST.

```
CONTROL PROMPT
   PROFILE MSGID
   XDEL NEVADA V(PRIM00)
   DATA PROMPT
   YES
   ENDDATA
```

Note: You do not need to include the SETPROMPT and PUSH statements if you specify the NOVERIFY option.

For additional information about the XDELPAIR command and how to use XRC utility devices, refer to “XDELPAIR–Deleting volume pairs” on page 82 and “Using XRC utility devices” on page 165.

Using the XSUSPEND command

The following section includes information suspending XRC volume pairs and XRC sessions.

Suspending XRC volume pairs

Whenever volume pairs are in a suspended state, XRC does not apply primary volume updates to the suspended secondary volumes. A track-level bitmap is maintained of changed tracks that are associated with the suspended primary volumes. Suspension messages report the consistency group time for the suspended volumes.

The suspension of a volume pair, as the result of an error, may cause XRC to take action on other volume pairs as a result. The ERRORLEVEL that you specified for the suspended volume pair, and the scope of the error, both determine the actions that are taken by XRC.

If errors occur while a volume pair is in suspended state, the pair remains in a suspended state and the ERRORLEVEL parameter has no effect on suspended volume pairs. Certain severe errors such as loss of the software or hardware bitmap records may require that you do a full-volume resynchronization on the volume pair when you add it back to the XRC session.

For additional information about the XSUSPEND command and associated parameters, refer to “XSUSPEND–Suspending volumes or sessions” on page 114.

For additional information about each ERRORLEVEL specification, refer to Chapter 11, “Recovering from error conditions using extended remote copy,” on page 255.

Suspending an XRC session

For storage controls that support hardware bitmaps, suspension of an XRC session has no application performance impact. For storage controls that do not support hardware bitmaps, a suspended XRC session can affect the system environment. The actual impact depends on the configuration, the activity level of the volume pairs in the suspended session, and the primary storage control's timeout value.

The capability of the storage control largely determines what happens when an XRC session becomes suspended. Storage controls that are able to do so manage record updates with a hardware bitmap. For these types of volumes, the outage (planned or unplanned) can be of any duration. The primary impact is that the resynchronization time can be longer for an extended outage.

Managing cache resources on storage subsystems that support hardware bitmaps

When XRC suspends a storage control session on a storage subsystem, such as an ESS, that supports hardware bitmaps, the storage control does not need to store changes in its cache storage. The hardware maintains a bitmap of primary system updates.

Managing cache resources on storage subsystems that do not support hardware bitmaps

When XRC suspends a storage control session on a storage control that does *not* support hardware bitmaps, the storage control continues to accumulate changes in its cache storage. At the end of the TIMEOUT duration, the storage control ends the storage control session. When you restart the XRC session with an XSTART command, the system data mover drains the storage control cache and accumulates the updates in a software bitmap. When you issue an XADDPAIR command, the software uses the software bitmap to resynchronize the volumes in the XRC session.

A suspended storage control session that experiences a lot of update activity places a greater demand on existing cache resources. This suspended session has an increased potential to significantly affect primary system performance. The storage control may cancel storage control sessions as a result.

If you do not issue the XSTART command before the timeout expires, the storage control *ends* the XRC session. You can monitor the current timeout for a storage control by issuing an “F ANTAS000,SCDATA” command. If the timeout is close to expiration, follow the procedure that is described in “Examples: changing TIMEOUT intervals for suspended sessions” on page 202 to avoid having to recopy the volumes on the storage control.

If a suspended XRC session requires significant cache resources, other applications that are accessing the same storage control can encounter IEA482I, IEF196I, or ADR348E messages. You may choose to cancel the jobs that generate the messages, then rerun them after you have restarted the XRC session. Be aware that DFSMSdss concurrent copy sessions and suspended XRC sessions both use cache resources. The potential exists to overrun these resources, resulting in IEA482I, IEF196I, or ADR348E messages.

During any planned outage, primary systems that issue RESERVE commands to the utility volume will enter into a long-busy condition. Either a storage control report or the "F ANTAS000,SCDATA" command can identify the utility volumes in a suspended storage control session.

For additional information about an SCDATA operation, refer to “SCDATA operation (XRC, CC)” on page 557.

Avoiding deadlock conditions

XRC allocates the state, control, and journal data sets when you issue an XSTART command. The catalog that is accessed by the HLQ keyword on the XSTART command controls access to these data sets. Ensure that application programs have not reserved this catalog data set, as this can cause a long busy condition (with resultant IEA482I or IEF196I messages). If this occurs, the XSTART function fails with a return code of 435 or 486 when attempting to allocate these data sets. You will need to cancel the jobs causing the deadlock situation.

You may avoid this deadlock condition by having the XRC state, control, and journal data sets in a user catalog that contains only entries for these data sets. Also ensure that either XRC is not copying the volume these data sets are on, or the volume does not contain data sets that application programs are updating. Either of these conditions can cause a deadlock condition.

Adding back suspended volumes to the XRC session

Issue the XADDPAIR command to add individual volume pairs, all suspended volume pairs, or suspended utility volumes back into an XRC session.

For additional information about the XADDPAIR command and the associated parameters, refer to “XADDPAIR–Adding volume pairs or utility volumes” on page 71.

Adding back specific suspended volume pairs

Issue an XADDPAIR command to add specific suspended volume pairs back to the XRC session. An XADDPAIR command that is issued to an active volume pair generates an error message. The XADDPAIR command resynchronizes each volume pair by copying all tracks of data that have changed since the time of suspension. If the volume pair was in pending state, the initialization copy continues from the point where the pair became suspended. It also copies all data that has changed since the pair became suspended.

You can change the keyword options when you reissue an XADDPAIR command to a suspended volume pair, with the following constraints:

- You cannot change the primary and secondary volumes in the pair. They must be the same volumes that you initially specified.
- You may change the ERRORLEVEL value. Suspended volumes will keep the same characteristics that existed when the suspension occurred.

- You may change the DVCBLOCK attributes by either specifying DONOTBLOCK or DVCBLOCK. If you do not explicitly specify DONOTBLOCK or DVCBLOCK and have not changed the DVCBLOCKing characteristics using XSET, then the DVCBLOCKING characteristics specified before suspension will be used when re-added.

  Note: DVCBLOCK and DONOTBLOCK are affected by workload-based write pacing. For more information, refer to "Workload-based write pacing" on page 161.
- You may add the MSGROUTEID keyword, or specify a new routing.
- You cannot change the SCSESSION value.
- XRC will ignore the NOCOPY, FULLCOPY, or QUICKCOPY keyword.

Adding back all suspended volume pairs

Resynchronize **all** of the suspended volumes that are in an XRC session with the SUSPENDED parameter of the XADDPAIR command. The volume pair keeps the same characteristics that you specified with the XADDPAIR command that created the volume pairs.

The SUSPENDED parameter provides a rapid way to restore the environment as it existed prior to either a session-level suspension or the suspension of all volumes in the session. However, you cannot use this option to change the characteristics (such as error level and session assignment) of a volume pair.

Adding a suspended utility volume back to the storage control session

If an error causes the primary storage control session to end, the SDM places the utility volume or volumes in a suspended state. These suspended utility volumes are then not accessible for use.

After you fix the problem that caused the error, reissue an XADDPAIR command to reactivate suspended utility volumes and add them back into the storage control session.

Using XRC with FlashCopy and tertiary volumes

The use of ESS at the recovery site provides a unique ability to combine FlashCopy functions with XRC functions. You can use this combination to create a consistent copy of data for disaster recovery testing or point-in-time backup.

The following scenario describes a method of creating a usable tertiary copy of data while keeping the existing data mover session active:

1. Suspend all volume pairs in the XRC session using the XSUSPEND VOLUME(ALL) command.
2. Issue FCESTABL commands to FlashCopy the XRC secondary volumes to tertiary volumes within the same ESS. Do not include volumes that contain the journal, state, control, and master data sets.
3. Use the DFSMDdss COPY DATASET command to FlashCopy XRC journal, state, control, and master data sets to tertiary volumes within the same ESS. Use the RENAMEUNCONDITIONAL parameter, specifying high level qualifiers different from those in use by the currently active session. Also specify the TOLERATE(ENQFAILURE) parameter.

4. Once the FlashCopy relationships are established, issue the XADDPAIR command with the SUSPENDED parameter to begin resynchronization of the active session.
5. Vary the secondary volumes (FlashCopy source volumes) offline to the system on which the XRECOVER command will be issued.
6. Vary the tertiary volumes online to the system on which the XRECOVER command will be issued.
7. Issue the XRECOVER command with the HLQ and MHLQ parameters specifying the new high level qualifiers used in step 3 on page 182, and either the ONLINE or TERTIARY parameter. The ONLINE or TERTIARY parameter indicates that it is permissible to perform the recovery to volumes other than the original secondaries. ONLINE indicates that any volumes matching the volsers of the original secondaries may be used; TERTIARY specifies that the secondaries that are used must not be the same as those that were in use when the XSUSPEND was issued. The HLQ parameter is used to locate the XRC control, state, and journal data sets. The MHLQ parameter is used to locate the XRC master data set copied in step 3 on page 182.

   Note: The XRECOVER command cannot run on the system that is currently running the active session. Any attempt to do so will fail with a return code 416. The command may be issued on a different system running in the same SYSPLEX.

Using the XSET TIMEOUT parameter

The XSET TIMEOUT keyword specifies the maximum application impact that is allowed by an XRC session. As a result of the TIMEOUT value, a storage control that supports hardware bitmapping automatically *suspends* an XRC session if the SDM does not read from the storage control sidefile within the timeout period. This action frees up storage control resources. The hardware bitmap and application programs continue with no performance impact. XRC resynchronizes the volumes, based on the hardware bitmap, when it receives a subsequent XADDPAIR command for the volumes on that storage control. The volumes are not consistent with the session consistency group time until the resynchronization is complete.

Note: The effect of the TIMEOUT option on the license internal code (LIC) operation for the disk subsystem is specific to the microcode implementation for that subsystem. When you are using non-IBM primary disk subsystems, consult the vendor of the subsystem to determine an optimum value.

A storage control that does not support hardware bitmapping *ends* the storage control session if the timeout interval expires. This frees all storage control resources so that there is no application performance impact. The fact that the storage control session has ended, however, means that you must completely synchronize the volumes in the XRC session from scratch. The volumes are not consistent with the XRC session consistency group time until the resynchronization is complete.

If you set the XSET TIMEOUT value low, protection is provided for applications at the expense of disaster recovery protection because session suspension or termination requires volume resynchronization to restore disaster protection. After the suspension or termination is done, all data on the secondary volumes is consistent with the reported timestamp.

If you set the XSET TIMEOUT value high, an XRC environment may avoid being suspended, but application slowdown might be experienced if the SDM stops reading updates from the primary subsystems, until the specified time has expired.

Recommendation: For IBM storage subsystems, it is recommended that you not set the XSET TIMEOUT value lower than the missing interrupt handler (MIH) value for the primary volumes and never set it lower than the absolute minimum, 20 seconds. The default value is 5 minutes. The TIMEOUT interval does not begin until the residual in the primary storage control exceeds a high threshold value. You can minimize impact to application programs with the TIMEOUT value, application update rate, and the storage control cache size. Most customers specify the TIMEOUT value on the XSET command in the 30–90 second range.

Note: When using channel extenders, setting the XSET TIMEOUT to the same value as the missing interrupt handler (MIH) is the norm.

Running the ICKDSF program

Table 30 contains information about running the ICKDSF REFORMAT REFVTOC program. Note that this function should be run from a system in the production SMSPlex to ensure that DFSMS space statistics are correctly updated and accurately reflect the space on the primary and/or secondary volumes.

Table 30. Determining when to run the ICKDSF REFORMAT REFVTOC program

If . . .	Then . . .	Comments . . .
An XRC volume pair is established or reestablished using the NOCOPY option	Run the ICKDSF REFORMAT REFVTOC program on the primary volume after the pair is established.	The volume table of contents (VTOC) and indexed VTOC information then reflect the primary volume that is being accessed.
The primary and secondary of this pair is switched at any time	Run the ICKDSF REFORMAT REFVTOC program on the new primary volume	The VTOC and indexed VTOC information then reflect the new primary volume.
The secondary volume is removed from the pair for use as a simplex volume	Run the ICKDSF REFORMAT REFVTOC program on the volume	The VTOC and indexed VTOC information then reflect the correct information for the device.

Using XRC reports

Use the XQUERY command to query the status of an active XRC session. The XQUERY command can generate the following reports:
- Session summary
- Volume summary
- Settings for session defaults
- Storage control activity
- Detail of volume activity
- Configuration activity
- Master session summary (in coupled environments only)
- SWAP

CXRC enhances XQUERY reports which return a consistency time to report a recoverable delay. For a coupled session, the recovery delay is the difference between the XRC session's latest controller time and the master session's journal time. For an uncoupled session, the recovery delay is the difference between the latest time reported by a controller session under that XRC session and the journal time of the session. For the duration of this delay, XRECOVER cannot recover applications. The recovery delay can be thought of as the degree of exposure of data to a disaster.

Note:

1. For a configuration report or volume report, you can use combinations of the filtering keywords. The report will include the configurations (or volumes) that meet *all* of the specified filters. See the "XQUERY configuration report" on page 188 and "XQUERY volume report" on page 186 for details.
2. The order of the volume pairs in an XQUERY report might differ from the order in which the volume pairs were added to the XRC session.
3. "Recovering data with XRC—Uncoupled and coupled sessions" on page 246 describes a special XQUERY report that the XRECOVER command automatically generates as part of a recovery session.
4. XRC automatically generates an XQUERY storage control report when an XSUSPEND command suspends an XRC session.
5. If you direct XQUERY command output to a data set, XRC does not also provide the output to the system's log. Remember that when you specify the data set name, standard TSO conventions apply. Unless you enclose the data set name in quotation marks, TSO appends the user ID to the front of the data set name.
6. XQUERY command output in the SYSLOG can contain a connection code at the end of the ANTQ8200I message and at the beginning of each of the lines associated with that connection code. The system generates this code for any multiline WTO.
7. XRC automatically generates an XQUERY MASTER report during an XEND, XSUSPEND TIMEOUT(), or XADVANCE of a coupled session.

For additional information about the XQUERY commands that produce the reports, refer to "XQUERY–Querying a session" on page 90. *z/OS MVS System Messages, Vol 1 (ABA-AOM)* contains details about the XRC messages in the reports.

Examples of XQUERY reports in an uncoupled environment

The reports in this section are examples of XQUERY reports that are generated in an uncoupled environment.

XQUERY summary report

The following is an example of an XQUERY summary report that summarizes the operating parameters for the XRC session. This report was generated by the XQUERY SESSION1 command.

```
ANTQ8200I XQUERY STARTED FOR SESSION(SESSION1) ASNAME(ANTAS001)
ANTQ8202I XQUERY SUMMARY REPORT - 002
ANTQ8203I -------------------------------------------------------------
ANTQ8238I TOTAL=37  DUP=32  CPY=1  PND=1  SUS=2  SEQ=0  UTL=1
ANTQ8260I PAV=0
ANTQ8231I DATA CONSISTENT(2003.266 19:41:27.998450)  DELAY(00:00:01.83)
ANTQ8240I DATA EXPOSURE(00:00:00.68)
ANTQ8232I SESSIONTYPE(XRC)  ERRORLEVEL(VOLUME)  HLQ(SYS1)
ANTQ8233I DEFAULT TIMEOUT(STORAGE_CONTROL_DEFAULT)
ANTQ8234I PND SESSION CMD = XSUSPEND ATTIME(2003.267 12:00:00.000000)
ANTQ8236I PND XSUSPEND TIMEOUT(STORAGE_CONTROL_DEFAULT)
ANTQ8201I XQUERY SUMMARY REPORT COMPLETE FOR SESSION(SESSION1)
```

XQUERY volume report

The following is an example of an XQUERY volume report. This report provides a complete volume summary and is similar to the report that the initial remote copy release provided. This report was generated by the XQUERY SESSION1 VOLUME(ALL) command.

```
ANTQ8200I XQUERY STARTED FOR SESSION(SESSION1) ASNAME(ANTAS001)
ANTQ8202I XQUERY VOLUME REPORT - 003
ANTQ8211I PRIM   SEC     ERROR  SYNCH
ANTQ8212I VOL    VOL     LEVEL    % STA CMD OP -------TIMESTAMP--------
ANTQ8203I -------------------------------------------------------------
ANTQ8213I PRIM01 SECD01 SESSION     DUP
ANTQ8213I PRIM02 SECD02 VOLUME   21 CPY
ANTQ8213I PRIM03 SECD03 SESSION     SEQ
ANTQ8213I PRIM04 SECD04 VOLUME   95 SUS        2004.135 11:59:43.583528
ANTQ8213I PRIM05 SECD05 SESSION  90 SUS        2004.135 12:48:48.071334
ANTQ8213I PRIM06 SECD06 VOLUME      DUP DEL DR 2004.135 12:56:58.276463
ANTQ8213I PRIM07 SECD07 VOLUME      DUP
ANTQ8213I PRIM08 SECD08 VOLUME      DUP SUS AT 2004.135 12:49:25.984596
ANTQ8213I PRIM09 SECD09 VOLUME      DUP SUS DR 2004.135 12:59:28.048573
ANTQ8213I PRIM10 SECD10 VOLUME      DUP
ANTQ8213I PRIM11 SECD11 IMSGROUP 95 CPY DEL AT 2004.135 12:38:18.771624
ANTQ8213I PRIM12 SECD12 IMSGROUP 00 PND DEL DR 2004.135 12:58:21.771624
ANTQ8213I PRIM13 SECD13 VOLUME   57 CPY
ANTQ8213I PRIM14 SECD14 DB2GROUP 00 PND SUS AT 2004.135 12:58:30.894533
ANTQ8213I PRIM15 SECD15 DB2GROUP 83 CPY SUS DR 2004.135 12:58:38.781519
ANTQ8213I PRIM16 SECD16 VOLUME   00 PND
ANTQ8213I PRIM17 SECD17 VOLUME      SEQ SUS AT 2004.135 12:58:30.894533
ANTQ8213I PRIM18 SECD18 GROUP1      SEQ SUS DR 2004.135 12:58:38.781519
ANTQ8213I PRIM19 SECD19 VOLUME      SEQ
ANTQ8213I PRIM20 SECD20 VOLUME   ** SUS        2004.135 12:00:45.936853
ANTQ8213I PRIM21 SECD21 VOLUME   42 SUS        NO_TIME_AVAILABLE
ANTQ8213I PRIM22 XRCUTL          -- UTL
ANTQ8213I PRIM23                 -- NIS
ANTQ8213I PRIM24 XRCUTL          -- UTL
ANTQ8238I TOTAL=23  DUP=6   CPY=4  PND=3  SUS=4  SEQ=4  UTL=2
ANTQ8260I PAV=0
ANTQ8231I DATA CONSISTENT(2004.135 12:48:27.998450)  DELAY(00:00:00.82)
ANTQ8240I DATA EXPOSURE(00:00:00.56)
ANTQ8232I SESSIONTYPE(XRC)  ERRORLEVEL(VOLUME)  HLQ(SYS1)
ANTQ8233I DEFAULT TIMEOUT(STORAGE_CONTROL_DEFAULT)
ANTQ8201I XQUERY VOLUME REPORT COMPLETE FOR SESSION(SESSION1)
```

XQUERY volume pace report

The following report is an example of an XQUERY volume pace report that is generated by the XQUERY S1 V(ALL) PACE command.

```
ANTQ8200I XQUERY STARTED FOR SESSION(S1) ASNAME(ANTAS001)
ANTQ8202I XQUERY VOLUME_PACE REPORT - 002
ANTQ8219I PRIM   SEC             SC SC RES      THD    RES  WRT  PACE
ANTQ8220I VOL    VOL    STA SSID SN ID CNT      CNT    RATE RATE MS
ANTQ8203I -------------------------------------------------------------
ANTQ8221I PRIA03 SECA03 DUP B100 -- 01 0003E45 *1400:5 0260 0260 0000.5
ANTQ8221I PRIA01 SECA01 DUP B100 -- 01 0006CEA  0000   0443 0443
ANTQ8221I PRIB01 SECB01 DUP A870 -- 01 0005DBB  0000
ANTQ8221I PRIB02 SECB02 DUP A870 -- 01 00004FF  0500
ANTQ8221I PRIA02 SECA02 DUP B100 -- 01 00004CC  0500   0026 0026
ANTQ8221I UTLA01 XRCUTL UTL B100 -- 01 0000000  0000   0000 0000
ANTQ8221I UTLB01 XRCUTL UTL A870 -- 01 0000000  0000
ANTQ8238I TOTAL=7  DUP=5  CPY=0  PND=0  SUS=0  SEQ=0  UTL=2
ANTQ8260I PAV=0
ANTQ8208I MONITOR INTERVAL STATISTICS AS OF 2005.307 04:27:14 UTC
ANTQ8201I XQUERY VOLUME_PACE REPORT COMPLETE FOR SESSION(S1)
```

Workload-based write pacing

When workload-based write pacing is in effect, the XQUERY volume pace report has these differences:

- It is labeled as REPORT 003, rather than REPORT 002
- Lines are added for the statistics of the levels of the write activity within the volumes.
- The values for RES RATE, WRT RATE, and PACE MS on the detail lines are the total values for the volume.
- When workload-based write pacing is in effect for a storage controller, the threshold count (THD CNT) column separator is changed from `:` to `@`, and the value after this separator is the setting for the write pacing level from an XADDPAIR or XSET command, not necessarily the level at which the volume is being paced.
- An additional detail line is shown for each WLM pacing level (WLM1-WLM6) that identifies what WLM importance levels are getting written to the volume, and the differentiated pacing in effect.

Note: With the additional information for workload-based write pacing, the number of lines in the XQUERY VOLUME PACE report may increase significantly.

For more information, refer to "Workload-based write pacing" on page 161.

XQUERY storage control detail report

The following is an example of an XQUERY storage control detail report for a session. This report was generated by a XQUERY TEXAS STORAGECONTROL DETAIL command. It includes information about the primary and auxiliary enhanced reader utility device addresses as denoted by p and a respectively.

```
ANTQ8200I XQUERY STARTED FOR SESSION(TEXAS) ASNAME(ANTAS001) 680
ANTQ8202I XQUERY STORAGECONTROL_DETAIL REPORT - 001
ANTQ8241I SC SC S RES UTIL CURRENT
ANTQ8242I SSID SN ID T CNT VOL TIMEOUT --STORAGE CONTROL TIME--
ANTQ8203I -------------------------------------------------------------
ANTQ8243I B8A0 AA 04 000000EaUAN001 00.05.00 2009.156 18:06:53.503070
ANTQ8243I B8A0 AA 05 000003AaUAN002 00.05.00 2009.156 18:06:53.503070
ANTQ8243I B8A0 AA 06 000000FaUAN002 00.05.00 2009.156 18:06:53.503070
ANTQ8243I B8A0 AA 03 0000000pUAN001 00.05.00 2009.156 18:06:53.503070
ANTQ8243I C8A0 BB 04 0000000aUBN001 00.05.00 2009.156 18:06:45.015218
ANTQ8243I C8A0 BB 05 0000000aUBN001 00.05.00 2009.156 18:06:45.015218
ANTQ8243I C8A0 BB 06 0000000aUBN001 00.05.00 2009.156 18:06:45.015218
ANTQ8243I C8A0 BB 03 0000000pUBN001 00.05.00 2009.156 18:06:45.015218
ANTQ8238I TOTAL=11 DUP=0 CPY=4 PND=4 SUS=0 SEQ=0 UTL=3
ANTQ8260I PAV=5
ANTQ8231I DATA CONSISTENT(2009.156 18:06:51.218619) DELAY(00:00:02.28)
ANTQ8201I XQUERY STO_DETAIL REPORT COMPLETE FOR SESSION(TEXAS)
```

XQUERY storage control XFEATURES report

The following is an example of a report generated by a XQUERY STORAGECONTROL command with the XFEATURE parameter. The report includes information about workload-based write pacing. Several fields are highlighted and are described further following the example.

```
ANTQ8202I XQUERY STORAGECONTROL_XFEATURES REPORT - 001
ANTQ8370I       T  LIC          FEATURES
ANTQ8371I SSID  Y  LEVEL        FU D W SL ER IR EX  WP
ANTQ8203I ----------------------------------------------------------
ANTQ8372I 7B74 U 5.3.1.172    YY N N NN YN Y  ED YY
ANTQ8372I 7B76 S 5.3.1.172    C* C C C* C* C  ** C*
ANTQ8375I UTL=1 SUTL=1
ANTQ8231I DATA CONSISTENT(NO_TIME_AVAILABLE) IDLE(00:00:22.6)
ANTQ8201I XQUERY STO_XFEATURES REPORT COMPLETE FOR SESSION(A)
```

The information related to workload-based write pacing is as follows:

- TY shows the type of utility: U for standard utility or S for swap utility.
- WP is the workload-based writing pacing status.
  - The fist character is the microcode status: - if the microcode support is not loaded into the storage controller and Y if the microcode support has been activated.
  - The second character is the software status: N if parmlib parameter WorkloadWritePacing is not active, and Y if parmlib parameter WorkloadWritePacing parameter is active.
- Message ANTQ8375I shows the total number of utilities and swap utilities in the session.
- Message ANTQ8372I, the swap utility detail line, shows these values:
  - For hardware features: C if it is capable and - if it is not installed.
  - For software features: * (XRC is not managing the devices).

XQUERY configuration report

The following is an example of an XQUERY configuration report that was generated by the XQUERY SESSION1 CONFIGURATION command.

```
ANTQ8200I XQUERY STARTED FOR SESSION(SESSION1) ASNAME(ANTAS001)
ANTQ8202I XQUERY CONFIGURATION REPORT - 003
ANTQ8281I ---------PRIMARY----------  -----SECONDARY------
ANTQ8284I      SC SC
ANTQ8282I SSID SN ID DVCN CCA VOLSER  SSID DVCN CCA VOLSER
ANTQ8203I ---------------------------------------------------------------
ANTQ8283I A760 AA 04 0F40 04  PRIM01  A764 0F50 03  SECD01
ANTQ8283I A760 AA 04 0F41 08  PRIM02  A764 0F51 0E  SECD02
ANTQ8283I 1010 AA 01 0F60 02  PRIM03  A764 0F52 01  SECD03
ANTQ8238I TOTAL=3  DUP=3  CPY=0  PND=0  SUS=0  SEQ=0  UTL=0
ANTQ8260I PAV=0
ANTQ8231I DATA CONSISTENT(2004.018 16:41:15.390719)   IDLE(00:40:06.3)
ANTQ8240I DATA EXPOSURE(00:00:00.00)
ANTQ8201I XQUERY CONFIGURATION REPORT COMPLETE FOR SESSION(SESSION1)
```

XQUERY SWAP report

The following is an example of an XQUERY SWAP report generated by an XQUERY S5 SWAP command. It includes enhanced reader information about the primary and auxiliary utility device addresses as denoted by p and a respectively.

```
ANTL8800I XQUERY S5 SWAP
ANTQ8200I XQUERY STARTED FOR SESSION(S5) ASNAME(ANTAS001) 502
ANTQ8202I XQUERY SWAP REPORT - 001
ANTQ8286I ---------PRIMARY-----------  ---SWAP---
ANTQ8284I      SC SC
ANTQ8287I SSID SN ID  DVCN CCA VOLSER  SSID  DVCN STA  SWP%   AGE
ANTQ8203I ---------------------------------------------------------
ANTQ8288I 79DC AA 01p D00C  3B MYD00C  7DDC  D00E UTL
ANTQ8288I 79DC AA 04a D00C  3B MYD00C  7DDC  D00E PAV
ANTQ8288I 79DC AA 03a D00D  3C MYD00D  7DDC  D00F UTL
ANTQ8288I 79DC AA 05a D00D  3C MYD00D  7DDC  D00F PAV
ANTQ8288I 79DC AA 01  1F41  24 MY1F41  7DDC  2F41 DUP    0  00:00:00
ANTQ8288I 79DC AA 01  1F43  26 MY1F43  7DDC  2F43 DUP    0  00:00:00
ANTQ8288I 79DC AA 01  1F44  27 MY1F44  7DDC  2F44 DUP    0  00:00:00
ANTQ8238I TOTAL=5  DUP=3  CPY=0  PND=0  SUS=0  SEQ=0  UTL=2
ANTQ8260I PAV=2
ANTQ8231I DATA CONSISTENT(NO_TIME_AVAILABLE)   IDLE(00:11:40.8)
ANTQ8240I DATA EXPOSURE(NO_TIME_AVAILABLE)
ANTQ8290I XRC OPERATING MODE(HYPER-PPRC)
ANTQ8201I XQUERY SWAP REPORT COMPLETE FOR SESSION(S5)
```

Examples of XQUERY reports in a coupled environment

The reports in this section are examples of XQUERY reports that are generated in a coupled environment. Use the name of the master session when you enter the command to generate the report. *z/OS MVS System Messages, Vol 1 (ABA-AOM)* contains details about the XRC messages that are included in the report.

XQUERY summary report

The following is an example of an XQUERY summary report that summarizes the operating parameters for the XRC master session. This report was generated by the XQUERY SESSION1 command.

```
ANTQ8200I XQUERY STARTED FOR SESSION(SESSION1) ASNAME(ANTAS002)
ANTQ8202I XQUERY SUMMARY REPORT - 002
ANTQ8203I -------------------------------------------------------------
ANTQ8238I TOTAL=37  DUP=32  CPY=1  PND=1  SUS=2  SEQ=0  UTL=1
ANTQ8260I PAV=0
ANTQ8231I DATA CONSISTENT(2003.266 19:41:27.998450)  DELAY(00:00:01.83)
ANTQ8240I DATA EXPOSURE(00:00:00.68)
ANTQ8232I SESSIONTYPE(XRC)  ERRORLEVEL(VOLUME)  HLQ(SYS1)
ANTQ8233I DEFAULT TIMEOUT(STORAGE_CONTROL_DEFAULT)
ANTQ8246I MSESSION(MSESS1) MHLQ(SYS1) COUPLED(INTERLOCKED)
ANTQ8234I PND MASTER CMD = XSUSPEND ATTIME(2003.267 12:00:00.000000)
ANTQ8236I PND XSUSPEND TIMEOUT(STORAGE_CONTROL_DEFAULT)
ANTQ8201I XQUERY SUMMARY REPORT COMPLETE FOR SESSION(SESSION1)
```

XQUERY volume report

The following is an example of an XQUERY volume report. This report provides a complete volume summary of the status of all volumes in the master session. This report was generated by the XQUERY SESSION1 VOL(ALL) command.

```
ANTQ8200I XQUERY STARTED FOR SESSION(SESSION1) ASNAME(ANTAS001)
ANTQ8202I XQUERY VOLUME REPORT - 003
ANTQ8211I PRIM   SEC     ERROR  SYNCH
ANTQ8212I VOL    VOL     LEVEL    % STA CMD OP -------TIMESTAMP--------
ANTQ8203I -------------------------------------------------------------
ANTQ8213I PRIM01 SECD01 SESSION     DUP
ANTQ8213I PRIM02 SECD02 VOLUME   21 CPY
ANTQ8213I PRIM03 SECD03 SESSION     SEQ
ANTQ8213I PRIM04 SECD04 VOLUME   95 SUS        2004.135 11:59:43.583528
ANTQ8213I PRIM05 SECD05 SESSION  90 SUS        2004.135 12:48:48.071334
ANTQ8213I PRIM06 SECD06 VOLUME      DUP DEL DR 2004.135 12:56:58.276463
ANTQ8213I PRIM07 SECD07 VOLUME      DUP
ANTQ8213I PRIM08 SECD08 VOLUME      DUP SUS AT 2004.135 12:49:25.984596
ANTQ8213I PRIM09 SECD09 VOLUME      DUP SUS DR 2004.135 12:59:28.048573
ANTQ8213I PRIM10 SECD10 VOLUME      DUP
ANTQ8213I PRIM11 SECD11 IMSGROUP 95 CPY DEL AT 2004.135 12:38:18.771624
ANTQ8213I PRIM12 SECD12 IMSGROUP 00 PND DEL DR 2004.135 12:58:21.771624
ANTQ8213I PRIM13 SECD13 VOLUME   57 CPY
ANTQ8213I PRIM14 SECD14 DB2GROUP 00 PND SUS AT 2004.135 12:58:30.894533
ANTQ8213I PRIM15 SECD15 DB2GROUP 83 CPY SUS DR 2004.135 12:58:38.781519
ANTQ8213I PRIM16 SECD16 VOLUME   00 PND
ANTQ8213I PRIM17 SECD17 VOLUME      SEQ SUS AT 2004.135 12:58:30.894533
ANTQ8213I PRIM18 SECD18 GROUP1      SEQ SUS DR 2004.135 12:58:38.781519
ANTQ8213I PRIM19 SECD19 VOLUME      SEQ
ANTQ8213I PRIM20 SECD20 VOLUME   ** SUS        2004.135 12:00:45.936853
ANTQ8213I PRIM21 SECD21 VOLUME   42 SUS        NO_TIME_AVAILABLE
ANTQ8213I PRIM22 XRCUTL          -- UTL
ANTQ8213I PRIM23                 -- NIS
ANTQ8213I PRIM24 XRCUTL          -- UTL
ANTQ8238I TOTAL=23  DUP=6   CPY=4  PND=3  SUS=4  SEQ=4  UTL=2
ANTQ8260I PAV=0
ANTQ8231I DATA CONSISTENT(2004.135 12:48:27.998450)  DELAY(00:00:00.82)
ANTQ8240I DATA EXPOSURE(00:00:00.56)
ANTQ8232I SESSIONTYPE(XRC)  ERRORLEVEL(VOLUME)  HLQ(SYS1)
ANTQ8233I DEFAULT TIMEOUT(STORAGE_CONTROL_DEFAULT)
ANTQ8246I MSESSION(MSESS1) MHLQ(SYS1) COUPLED(INTERLOCKED)
ANTQ8201I XQUERY VOLUME REPORT COMPLETE FOR SESSION(SESSION1)
```

XQUERY volume detail report

The following is an example of an XQUERY volume detail report for a session in a master session that was generated by the XQUERY SESSION1 VOL(ALL) DETAIL command. *z/OS MVS System Messages, Vol 1 (ABA-AOM)* contains details about the XRC messages that are included in the report.

```
ANTQ8200I XQUERY STARTED FOR SESSION(SESSION1) ASNAME(ANTAS001)
ANTQ8202I XQUERY VOLUME_DETAIL REPORT - 002
ANTQ8216I PRIM   SEC     ERROR  SYNCH          RES  THD       SC SC
ANTQ8217I VOL    VOL     LEVEL    % STA CMD OP CNT  CNT  SSID SN ID
ANTQ8203I -------------------------------------------------------------
ANTQ8218I PRIM01 SECD01 SESSION     DUP        0089 0500 8700 -- 01
ANTQ8218I PRIM02 SECD02 SESSION     DUP        0499 0500 8702 L+ 01
ANTQ8218I PRIM03 SECD03 SESSION     DUP*       0620 0500 1010 AA 02
ANTQ8218I PRIM04 SECD04 SESSION  76 CPY        0056 0500 1011 -- 02
ANTQ8218I PRIM05 SECD05 SESSION  24 CPY*       05DB 0500 8702 BB 03
ANTQ8218I PRIM06 SECD06 SESSION     DUP        01AB 0500 A760 CC 04
ANTQ8218I PRIM07 SECD07 SESSION     SUS        0000 0500 A760 CC 04
ANTQ8218I UTIL02 XRCUTL             UTL        0000 0000 8702 L+ FF
ANTQ8238I TOTAL=7  DUP=4  CPY=2  PND=0  SUS=1  SEQ=0  UTL=1
ANTQ8260I PAV=0
ANTQ8231I DATA CONSISTENT(2003.325 23:20:49.504484)  DELAY(00:00:02.35)
ANTQ8240I DATA EXPOSURE(00:00:00.92)
ANTQ8232I SESSIONTYPE(XRC)  ERRORLEVEL(SESSION)  HLQ(HLQ1)
ANTQ8233I DEFAULT TIMEOUT(STORAGE_CONTROL_DEFAULT)
ANTQ8246I MSESSION(MSESS1) MHLQ(SYS1) COUPLED(NON-INTERLOCKED)
ANTQ8201I XQUERY VOLUME_DETAIL REPORT COMPLETE FOR SESSION(SESSION1)
```

XQUERY MASTER report

The XQUERY MASTER report is produced by ANTAS000 from the current contents of the master coupling data set. It may be issued on any system with access to the data set, and does not require an XRC session to be active on the system where it is issued.

The following is an example of an XQUERY MASTER report that was generated by the XQUERY MSESS1 MASTER MHLQ(SYS1) command. *z/OS MVS System Messages, Vol 1 (ABA-AOM)* contains details about the XRC messages that are included in the report.

```
ANTQ8300I XQUERY STARTED FOR MSESSION(MSESS1) MHLQ(SYS1)
ANTQ8202I XQUERY MASTER REPORT - 002
ANTQ8302I SESSION   STA VOL INT CMD   JOURNAL DELTA      RCV/ADV DELTA
ANTQ8303I -------------------------------------------------------------
ANTQ8304I SESSION1  ACT      Y       =00:00:00.000000  +00:00:01.429395
ANTQ8304I SESSION1  ACT      Y       =00:00:00.235324  +00:00:01.456778
ANTQ8304I SESSION1  ACT      Y       =00:00:00.567735  +00:00:01.568245
ANTQ8304I SESSION1  ACT      Y       =00:00:00.837235  +00:00:01.569867
ANTQ8304I SESSION1  ACT      Y       =00:00:01.252353  +00:00:00.429764
ANTQ8304I SESSION2  ACT      N       -00:00:17.527111
ANTQ8305I TOTAL=6  ACT=6  SUS=0  END=0  ARV=0  RCV=0  UNK=0
ANTQ8308I MSESSION RECOVERABLE TIME(2002.233 02:56:12.711994)
ANTQ8309I INTERLOCKED=5  NON-INTERLOCKED=1
ANTQ8301I XQUERY MASTER REPORT COMPLETE FOR MSESSION(MSESS1)
```

The following is another example of an XQUERY MASTER report that was generated by the XQUERY MSESS1 MASTER MHLQ(SYS1) command. *z/OS MVS System Messages, Vol 1 (ABA-AOM)* contains details about the XRC messages that are included in the report.

```
ANTQ8300I XQUERY STARTED FOR MSESSION(MSESS1) MHLQ(SYS1)
ANTQ8202I XQUERY MASTER REPORT - 002
ANTQ8302I SESSION   STA VOL INT CMD   JOURNAL DELTA      RCV/ADV DELTA
ANTQ8303I -------------------------------------------------------------
ANTQ8304I SESSION1  ACT ERR  Y       =00:00:00.000000  +00:00:02.327186
ANTQ8304I SESSION2  ACT SYM  Y       +00:00:01.331259  =00:00:00.000000
ANTQ8304I SESSION3  ACT SYM  N       -00:00:02.348653
ANTQ8304I SESSION4  SUS AVS  N
ANTQ8305I TOTAL=4  ACT=3  SUS=1  END=0  ARV=0  RCV=0  UNK=0
ANTQ8306I MSESSION STATUS=HOLD
ANTQ8308I MSESSION RECOVERABLE TIME(2002.233 02:56:12.711994)
ANTQ8309I INTERLOCKED=2  NON-INTERLOCKED=2
ANTQ8301I XQUERY MASTER REPORT COMPLETE FOR MSESSION(MSESS1)
```

The following is an example of an XQUERY MASTER report that was generated for a session with a session type (T) of Cluster (indicated by C in message ANTQ8304I).

```
ANTQ8300I XQUERY STARTED FOR MSESSION(HOLIDAYS) MHLQ(SYS1) 019
ANTQ8202I XQUERY MASTER REPORT - 003
ANTQ8302I SESSION  T STA VOL INT CMD  JOURNAL DELTA    RCV/ADV DELTA
ANTQ8303I -------------------------------------------------------------
ANTQ8304I SUMMER   C ACT      Y      =00:00:00.000000 +00:00:02.717446
ANTQ8304I WINTER   C ACT      Y      +00:00:03.513562 +00:00:02.716593
ANTQ8305I TOTAL=2  ACT=2  SUS=0  END=0  ARV=0  RCV=0  UNK=0
ANTQ8308I MSESSION RECOVERABLE TIME(2005.323 18:28:24.169419)
ANTQ8309I INTERLOCKED=2  NON-INTERLOCKED=0
ANTQ8301I XQUERY MASTER REPORT COMPLETE FOR MSESSION(HOLIDAYS)
```

Examples of XQUERY ADVANCE and XQUERY RECOVER reports

The reports shown below are examples of XADVANCE and XRECOVER reports. See "Creating a recovery volume report" on page 248 for more information on generating recovery volume reports.

The following is an example of an XQUERY ADVANCE report that was generated by the XADVANCE SESSION1 HLQ(SYS1) command.

```
ANTQ8200I XQUERY STARTED FOR SESSION(SESSION1) ASNAME(ANTAS001)
ANTQ8202I XQUERY ADVANCE REPORT - 001
ANTQ8272I PRIMARY  SECONDARY STA CON  ------TIMESTAMP---------
ANTQ8203I --------------------------------------------------------
ANTQ8273I PAS001   SES001    SUS YES  2003.322 20:55:00.356210
ANTQ8273I PAS002   SES002    SUS YES  2003.322 20:55:00.356210
ANTQ8273I PAS002   SES002    SUS NO   2003.322 19:53:00.324610
ANTQ8237I TOTAL=3   DUP=0   PND=0   SUS=3
ANTQ8231I DATA CONSISTENT(2003.322 20:55:00.356210)
ANTQ8232I SESSIONTYPE(XRC)  ERRORLEVEL(SESSION)  HLQ(SYS1)
ANTQ8201I XQUERY ADVANCE REPORT COMPLETE FOR SESSION(SESSION1)
```

The following is an example of an XQUERY RECOVER report that was generated by the XRECOVER SESSION1 HLQ(SYS1) command.

```
ANTQ8200I XQUERY STARTED FOR SESSION(SESSION1) ASNAME(ANTAS001)
ANTQ8202I XQUERY RECOVER REPORT - 002
ANTQ8271I ------ORIGINAL-------
ANTQ8274I PRIMARY  SECONDARY STA CON CLP  ------TIMESTAMP---------
ANTQ8203I --------------------------------------------------------
ANTQ8275I PRIM01   SECD01    DUP YES YES
ANTQ8275I PRIM02   SECD02    DUP YES YES
ANTQ8275I PRIM03   SECD03    SUS NO  YES  2000.322 19:54:00.562356
ANTQ8275I PRIM04   SECD04    PND NO  NO
ANTQ8237I TOTAL=4   DUP=2   PND=1   SUS=1
ANTQ8231I DATA CONSISTENT(2000.322 20:55:00.356210)
ANTQ8232I SESSIONTYPE(XRC)  ERRORLEVEL(SESSION)  HLQ(SYS1)
ANTQ8201I XQUERY RECOVER REPORT COMPLETE FOR SESSION(SESSION1)
```

Accessing secondary volumes while XRC is active

Applications can read secondary volumes while XRC continues to copy data to these volumes. However, you should consider the following when accessing secondary volumes from another system:

- The secondary volume serial number is different than the primary volume serial number. Access to the secondary volume must be by a different catalog or by explicit unit, volume, and serial number reference.
- Whenever the VTOC location on a volume changes, the new location must be updated in an in-storage pointer to the VTOC. XRC performs this refresh on the SDM system whenever it copies the primary volume VTOC pointer to the secondary volume. To refresh the pointer on other systems attached to the secondary volume, issue the console commands to vary the secondary volume offline and then online. Failure to do this can result in I/O errors when attempting to access the old VTOC.
- XRC continuously copies application updates to a secondary volume that remains in the XRC session. A program that reads from the secondary volume must be able to manage the fact that the data continues to change.

Note: VSAM volume data set (VVDS) information will reflect the primary volume, not the secondary volume.

Error conditions can occur during reads, however, due to the way that XRC handles secondary I/O functions. An example of this is an I/O error that results when the VTOC is updated before XRC has completely copied the data to the secondary volume. Applications that access secondary XRC volumes must be able to handle these temporary error situations. You can often resend the read I/O to clear these errors.

Ending an XRC session

Issue the XEND command to end the extended remote copy session prior to recovering a failed system. XRC generates a message that indicates the time of the last known consistent state of all volume pairs.

After you issue the XEND command, XRC prompts you (unless you have specified NOVERIFY) to confirm the request to end the session. You must specify either YES or NO. You can include the XEND command that is written in REXX. If you do, you should either specify NOVERIFY or place your response to the prompt in the data stack before you issue the command. In a batch environment, you can stack the response.

The following command ends the XRC session:

```
XEND DALLAS
```

The following command ends the XRC session and routes XRC messages to user ID E299000:

```
XEND DALLAS MSGROUTEID(E299000)
```

The XEND command stops all updates to secondary volumes after the completion of the target consistency group to that secondary volume. XRC sends messages that indicate that the XRC session has ended, and includes the consistency group timestamp for the last applied updates to the secondary volumes.

For additional information about the XEND command and the associated parameters, refer to “XEND–Ending a session” on page 86.

Applying XRC operational scenarios

The scenarios in this section describe possible ways to manage your XRC sessions to perform or accommodate various planned data management situations. XRC scenarios include preparing for maintenance, moving your data to another site, and changing session values. There are also scenarios for coupled XRC sessions, including coupling new or existing XRC sessions, restarting an inactive XRC coupled session after an error occurs, applying software maintenance in a coupled environment, resuming operations after a sympathetic suspension, and recovering from a disaster.

Before you begin: You need to be familiar with the commands used in this scenario. For more information, refer to Chapter 5, "Extended remote copy command descriptions," on page 69.

Example: applying maintenance

The following scenario uses XRC commands to suspend an XRC session in preparation for applying maintenance to the SDM system.

1. Start a session named TUCSON for disaster recovery with the state, control, and journal data sets under a high-level qualifier of NEW. Use the command:

```
XSTART TUCSON SESSIONTYPE(XRC) ERRORLEVEL(VOLUME) HLQ(NEW)
```

   Note: An error on any volume suspends the volume; other volumes continue to be copied.

2. Allocate 70 MB of fixed real storage to the system data mover.

```
XSET TUCSON PAGEFIX(70)
```

3. Add a volume pair to XRC session TUCSON.

```
XADDPAIR TUCSON VOLUME(PRIM01,SEC001) SCSESSION(AB)
```

   The error response level defaults to what is specified on the XSTART command. The volume is assigned to storage control session AB.

4. Add another volume pair to the XRC session, specifying the error response level as SESSION. The volume is assigned to the default storage control session "--".

```
XADDPAIR TUCSON VOLUME(PRIM02,SEC002) ERRORLEVEL(SESSION)
```

   Normal data processing continues.

5. Suspend the XRC session (XSUSPEND) and apply maintenance to the system. In time, maintenance needs to be applied to the SDM system. The maintenance, which may include IPLing the SDM system or refreshing SYS1.LINKLIB, is expected to take a maximum of six hours. If the maintenance is to be applied to the SDM modules in SYS1.LINKLIB, you must cancel the ANTAS000 address space after the refresh of SYS1.LINKLIB to have the maintenance take effect.

```
XSUSPEND TUCSON TIMEOUT(06.00.00)
```

   During the suspension, storage controls that support hardware bitmapping record application updates in their hardware bitmaps. Storage controls that do not support hardware bitmapping do not record application updates. Instead, updates accumulate in cache until the XSTART command function is complete.

6. Restart the TUCSON session for disaster recovery.

```
XSTART TUCSON SESSIONTYPE(XRC) ERRORLEVEL(VOLUME) HLQ(NEW)
```

7. Allocate 70 MB of fixed real storage to the system data mover.

```
XSET TUCSON PAGEFIX(70)
```

8. Add all suspended volume pairs back to the XRC session.

```
XADDPAIR TUCSON SUSPENDED
```

   All ERRORLEVEL and SCSESSION values default to what they were prior to the suspension. XRC resynchronizes all volume pairs.

Example: preparing for disaster

The following scenario uses XRC commands to support a disaster recovery. The scenario objective is to initialize a set of volumes, copy changes on a continuous basis, and recover to a known point in time after the disaster.

1. Start a session named TUCSON for disaster recovery with the state, control, and journal data sets under a high-level qualifier of NEW. Use the command:

```
XSTART TUCSON SESSIONTYPE(XRC) ERRORLEVEL(VOLUME) HLQ(NEW)
```

   Note: An error on any volume suspends the volume; other volumes continue to be copied.

2. Add a volume pair to the XRC session.

```
XADDPAIR TUCSON VOLUME(PRIM01,SEC001)
```

   The error response level defaults to what is specified on the XSTART command. XRC assigns the volume pair to the default storage control session "--".

3. Add another volume pair to the XRC session, specifying the error response level as SESSION.

```
XADDPAIR TUCSON VOLUME(PRIM02,SEC002) ERRORLEVEL(SESSION)
```

   Normal data processing continues.

4. Issue the following command at the recovery site when a disaster occurs:

```
XEND TUCSON XRECOVER TUCSON HLQ(NEW)
```

   The XRECOVER command applies in-process updates to the secondary volumes and clips the secondary volumes to the primary volume serial numbers. The recovery is then complete, with the secondary volumes consistent up to the time reported by the XRECOVER command.

Example: migrating data

The following scenario uses XRC commands to support migration mode. The scenario objective is to migrate a set of volumes from one site to another site, and ensure data consistency to a known point in time after the migration. The migration mode cannot be used for coupled sessions.

1. Start a session named TUCSON for data migration with the state data set under a high-level qualifier of NEW. Use the command:

```
XSTART TUCSON SESSIONTYPE(MIGRATE) ERRORLEVEL(VOLUME) HLQ(NEW)
```

   Note: An error on any volume suspends the volume; other volumes continue to be copied.
2. Add volume pairs to the session for migration.

```
XADDPAIR TUCSON VOLUME(PRIM01,SEC001 PRIM02,SEC002)
```

   The error response level defaults to what is specified on the XSTART command, which is VOLUME. (Normal data processing continues.)
3. Quiesce the prior to issuing the following command. After the application programs have been quiesced, issue the following command to ensure that all updates in the storage controls have been applied to the secondary volumes.

```
XSUSPEND TUCSON DRAIN TIMEOUT(03.30.00)
```

   The timeout duration of three hours and 30 minutes means that you plan to restart this session by an XSTART command before this timeout duration has passed.
4. Issue the RECOVER command to clip the secondary volumes.

```
XRECOVER TUCSON HLQ(NEW)
```

   The migration is now complete with the secondary volumes consistent up to the time reported by the XRECOVER command.

Example: switching system modes

The following scenario uses XRC to support migration of a data center from one site to another site. After the migration, the scenario shows how the XRC session mode can be changed to activate the SDM in a disaster recovery mode.

1. Start a session named TUCSON for data migration with the state data set under a high-level qualifier of NEW.

```
XSTART TUCSON SESSIONTYPE(MIGRATE) ERRORLEVEL(VOLUME) HLQ(NEW)
```

Note: An error on any volume suspends the volume; other volumes continue to be copied.

2. Add a volume pair to the session for migration.

```
XADDPAIR TUCSON VOLUME(PRIM01,SEC001)
```

The error response level defaults to what is specified on the XSTART command.

3. Add another volume pair to the XRC session, specifying the error response level as VOLUME.

```
XADDPAIR TUCSON VOLUME(PRIM02,SEC002) ERRORLEVEL(VOLUME)
```

Normal data processing continues.

4. Quiesce the application programs prior to issuing the following XSUSPEND command.

```
XSUSPEND TUCSON DRAIN TIMEOUT(03.30.00)
```

In this example, the DRAIN keyword ensures that all updates in the storage controls are applied to the secondary volumes. The timeout duration of three hours and 30 minutes means that you plan to restart this XRC session with an XSTART command before this timeout duration has passed.

During the suspension, storage controls that support hardware bitmapping record application updates in their hardware bitmaps. Storage controls that do not support hardware bitmapping do not record application updates. Instead, updates accumulate in cache until the XSTART command function is complete.

5. Restart the application programs and the session. Restart the session as a disaster recovery session. The XSTART command must complete before the storage control timeout expires, or else the XRC session is ended.

```
XSTART TUCSON SESSIONTYPE(XRC) ERRORLEVEL(VOLUME) HLQ(NEW)
```

An error on any volume causes that volume to be suspended; other volumes continue to be copied. The state, control, and journal data sets are under a high-level qualifier of NEW.

6. Resynchronize the volume pair in the session.

```
XADDPAIR TUCSON VOLUME(PRIM01,SEC001)
```

The error response level remains what was initially specified prior to the suspension.

7. Resynchronize the volume pair in the session.

```
XADDPAIR TUCSON VOLUME(PRIM02,SEC002) ERRORLEVEL(SESSION)
```

The error response level for this volume pair is changed to SESSION. (Normal data processing continues.)

8. Issue the following command when a disaster occurs. This command is issued at the recovery site to apply in-progress updates to the secondary volumes and clip the secondary volumes to the primary volume serial numbers.

```
XRECOVER TUCSON HLQ(NEW)
```

The recovery is complete with the secondary volumes consistent up to the time reported by the XRECOVER command.

Example: changing the session error recovery level

The following scenario uses XRC commands to change the XRC session error recovery level.

1. Start a session named TUCSON for disaster recovery.

```
XSTART TUCSON SESSIONTYPE(XRC) ERRORLEVEL(SESSION)
```

An error on any duplex volume causes the XRC session and all volumes in the session to be suspended. If an error occurs, all volumes in the session must be resynchronized. The state, control, and journal data sets are under a high-level qualifier of SYS1.

2. Add a volume pair to the XRC session, which inherits ERRORLEVEL(SESSION) from the XSTART command.

```
XADDPAIR TUCSON VOLUME(PRIM01,SEC001)
```

3. Add another volume pair to the XRC session, specifying the error recovery level as VOLUME.

```
XADDPAIR TUCSON VOLUME(PRIM02,SEC002) ERRORLEVEL(VOLUME)
```

Normal data processing continues.

4. Suspend the XRC session to the specific timestamp after all updates to the specified UTC timestamp have been applied to the secondary volumes.

```
XSUSPEND TUCSON ATTIME(1998.266 19:41:20.231440) TIMEOUT(03.30.00)
```

The timeout duration of three hours and 30 minutes means that you plan to restart this session by an XSTART command before this timeout duration has passed.

During the suspension, storage controls that support hardware bitmapping record application updates in their hardware bitmaps. Storage controls that do not support hardware bitmapping do not record application updates. Instead, updates accumulate in cache until the XSTART command function is complete.

5. Start a session named TUCSON for disaster recovery.

```
XSTART TUCSON SESSIONTYPE(XRC) ERRORLEVEL(VOLUME)
```

   The default error recovery level was SESSION. Specifying ERRORLEVEL(VOLUME) changes the default error recovery level to VOLUME for all new volumes that are added to the session. The state, control, and journal data sets are under a high-level qualifier of SYS1.
6. Resynchronize these suspended volume pairs.

```
XADDPAIR TUCSON VOLUME(PRIM01,SEC001,PRIM02,SEC002)
```

   Since this is a volume resynchronization, the volume pair keeps its original error recovery level of SESSION.
7. Add a volume pair to the session.

```
XADDPAIR TUCSON VOLUME(PRIM03,SEC003)
```

8. Add another volume pair to the session with an error recovery level of SESSION.

```
XADDPAIR TUCSON VOLUME(PRIM04,SEC004) ERRORLEVEL(SESSION)
```

   If an error occurs on this volume, the entire XRC session is suspended.

Example: changing a volume pair's error recovery level

The following scenario uses XRC commands to change the error recovery level for a volume pair, possibly because the volume is no longer critical to the XRC session.

1. Start a session named TUCSON for disaster recovery, using this command:

```
XSTART TUCSON SESSIONTYPE(XRC) ERRORLEVEL(SESSION)
```

   An error on any volume causes the XRC session and all volumes in the session to be suspended. All volumes must be resynchronized.
2. Add a volume pair to the XRC session.

```
XADDPAIR TUCSON VOLUME(PRIM01,SEC001)
```

   The volume inherits the SESSION error recovery level from the XSTART command. (Normal data processing continues.)
3. Suspend the volume.

```
XSUSPEND TUCSON VOLUME(PRIM01)
```

Storage controls that support hardware bitmapping record application updates in their hardware bitmaps. Storage controls that do not support hardware bitmapping record application updates in software bitmaps.

4. Resynchronize this suspended volume and change the error recovery level from SESSION to VOLUME.

```
XADDPAIR TUCSON VOLUME(PRIM01,SEC001) ERRORLEVEL(VOLUME)
```

Example: suspending and resynchronizing a volume pair

The following scenario describes the XRC commands that are used to suspend and resynchronize a volume pair. You might choose to do perform this scenario if a primary volume will receive many overlapping changes in a short time and, to optimize performance, and only want the final results to be copied.

1. Start a session named TUCSON for disaster recovery, using the command:

```
XSTART TUCSON SESSIONTYPE(XRC) ERRORLEVEL(SESSION)
```

Note: An error on any duplex volume will cause the XRC session and all duplex pairs in the session to be suspended. All volumes must be resynchronized. The state, control, and journal data sets are under a high-level qualifier of SYS1.

2. Add a volume pair to the XRC session, with session-level recovery from the XSTART command.

```
XADDPAIR TUCSON VOLUME(PRIM01,SEC001)
```

Normal data processing continues.

3. Suspend the volume.

```
XSUSPEND TUCSON VOLUME(PRIM01)
```

Storage controls that support hardware bitmapping record application updates in their hardware bitmaps. Storage controls that do not support hardware bitmapping record application updates in software bitmaps.

4. Resynchronize the suspended volume.

```
XADDPAIR TUCSON VOLUME(PRIM01,SEC001)
```

The state, control, and journal data sets are under a high-level qualifier of SYS1.

Example: suspending and resynchronizing a session

The following scenario describes XRC commands that are used for suspending and resynchronizing the XRC session to allow you to change to another virtual server or processor.

1. Start a session named TUCSON for disaster recovery, using the following command:

```
XSTART TUCSON SESSIONTYPE(XRC) ERRORLEVEL(SESSION)
```

   Note: An error on any volume will cause the XRC session and all volumes in the session to be suspended. All volumes must be resynchronized. The state, control, and journal data sets are under a high-level qualifier of SYS1.

2. Add the volume pair to the XRC session.

```
XADDPAIR TUCSON VOLUME(PRIM01,SEC001)
```

   The volume inherits session-level recovery from the XSTART command. (Normal data processing continues.)

3. Suspend the XRC session, and make the planned system changes.

```
XSUSPEND TUCSON TIMEOUT(03.30.00)
```

   The timeout duration of three hours and 30 minutes means that you plan to restart this session by an XSTART command before this timeout duration has passed.

   During the suspension, storage controls that support hardware bitmapping record application updates in their hardware bitmaps. Storage controls that do not support hardware bitmapping do not record application updates. Instead, updates accumulate in cache until the XSTART command function is complete.

4. Restart the XRC session on the new system.

```
XSTART TUCSON SESSIONTYPE(XRC) ERRORLEVEL(SESSION)
```

   The state, control, and journal data sets are under a high-level qualifier of SYS1.

5. Add back to the session all pairs that are in suspended state at the time that you initially issued the XADDPAIR command. Issue the following command:

```
XADDPAIR TUCSON SUSPENDED
```

   XRC starts volume synchronization processing, subject to the SYNCH and SCSYNCH values.

Example: changing volume channel connection addresses

When a volume is added to a session by the XADDPAIR command, the system data mover uses the volume serial number to determine which storage control the volume belongs to and which channel connection address to use to address the volume. If the channel connection address for an XRC volume pair is changed while the pair is still in an active XRC session, the SDM is unable to perform the copy operation. The following scenario describes how to register a changed channel connection address with the SDM.

Alerting the system data mover to a changed volume CCA

1. Issue the following command to add Volume PRIM01 to the XRC session on storage control A with a channel connection address of 123:
   (Normal data processing continues.)

```
XADDPAIR TUCSON VOLUME(PRIM01,SECD01)
```

2. Issue the following command to remove volume PRIM01 from the XRC session before the channel connection address is changed. As part of a planned configuration change, a service representative will move volume PRIM01 to storage control B with a new channel connection address of 456.

```
XDELPAIR TUCSON VOLUME(PRIM01)
```

3. Issue the following command to add Volume PRIM01 into the XRC session. At this point in time volume PRIM01 is on storage control B with the channel connection address of 456. You can add the volume to the session using the NOCOPY option if no application update activity has occurred during the switch-over.

```
XADDPAIR TUCSON VOLUME(PRIM01,SECD01)
```

Examples: changing TIMEOUT intervals for suspended sessions

When an XRC session is suspended, you must provide a TIMEOUT interval. The system data mover communicates this timeout value to all primary storage controls, and the session remains suspended for at least the specified timeout interval.

Changing XSUSPEND TIMEOUT intervals

The following scenario shows how to change a previously specified timeout interval. It may be that the planned outage is taking longer than you thought it would, or that it is causing unanticipated production impact. You have decided to suspend the XRC sessions.

1. Suspend the session named TUCSON. The planned outage is expected to be completed within 12 hours.

```
XSUSPEND TUCSON TIMEOUT(12.00.00)
```

   During the suspension, storage controls that support hardware bitmapping record application updates in their hardware bitmaps. Storage controls that do not support hardware bitmapping do not record application updates. Instead, updates accumulate in cache until the XSTART command function is complete.

(Normal data processing continues.) However, after about ten hours you realize that the timeout interval is not long enough.

2. Restart the session named TUCSON, causing the SDM to start recording changes to volumes in the session.

```
XSTART TUCSON SESSIONTYPE(XRC) ERRORLEVEL(VOLUME)
```

3. Issue the following command to start a new timeout period of five hours:

```
XSUSPEND TUCSON TIMEOUT(05.00.00)
```

   The XRC session remains suspended for up to five hours, which is plenty of time to complete the planned work.

Note: During the suspension, storage controls that support hardware bitmapping record application updates in their hardware bitmaps. Storage controls that do not support hardware bitmapping do not record application updates. Instead, updates accumulate in cache until the XSTART command function is complete.

Changing TIMEOUT for all SC sessions

You can change the TIMEOUT value for all volumes in an XRC session by issuing the XSET command. Specifying the TIMEOUT option, with the SSID(ALL) subparameter, changes the time for all storage controls. When a session is suspended, the current timeout value (the last value specified) remains in effect for the restarted sessions.

An XRC session called TUCSON is currently running with a timeout value set to the default of five minutes. You want to change the timeout value for all of the storage controls to be one minute.

1. Issue the following command so that all storage controls have a timeout value of one minute.

```
XSET TUCSON TIMEOUT(00.01.00)
```

   Some of the hardware that is related to the session requires planned maintenance that will take two hours. You need to suspend the session during this time. You have planned the suspension to happen during a time when application updates are minimal.

2. Issue the following command:

```
XSUSPEND TUCSON TIMEOUT(02.00.00)
```

   For 3990 or 9390 Storage Controls, if there are sufficient updates during this time it is possible that the storage control session can be canceled. For ESS storage subsystems, the hardware bitmap function keeps the storage control session active regardless of the timeout value that you have specified.

   Normal data processing continues.

3. Issue the following command to restart the session named TUCSON because the hardware updates have been completed:

```
XSTART TUCSON SESSIONTYPE(XRC) ERRORLEVEL(SESSION)
```

The SDM automatically sets the timeout value in the storage controls to the value specified by the last XSET command, which was TIMEOUT(00.01.00).

4. Issue the following command so that XRC can add all volumes back to the session and resynchronizes them using the hardware or software bitmaps.

```
XADDPAIR TUCSON SUSPENDED
```

Changing TIMEOUT for specific SC sessions

You can change the timeout interval for all volumes in an XRC session that belong to a specific SSID by issuing the XSET TIMEOUT command with the SSID subparameter. The timeout interval that you specify will remain in effect across session suspensions.

An XRC session called TUCSON is currently running with a timeout value set to the default of five minutes for all storage controls. You wish to change the timeout value for storage controls 10FC and 10E2 to be one minute.

1. Issue the following command so that the 10FC and 10E2 storage controls will have a timeout value of one minute.

```
XSET TUCSON TIMEOUT(00.01.00) SSID(10FC, 10E2)
```

All other storage controls continue to have a timeout value of five minutes. Some of the hardware that is related to the session requires planned maintenance that will take two hours. You need to suspend the session during this time. You have planned the suspension to happen during a time when application updates are minimal.

2. Issue the following command:

```
XSUSPEND TUCSON TIMEOUT(02.00.00)
```

For 3990 or 9390 Storage Controls, if there are sufficient updates during this time it is possible that the storage control session can be canceled. For ESS storage subsystems, the hardware bitmap function keeps the storage control session active regardless of the timeout value that you have specified.

Normal data processing continues.

3. Issue the following command to restart the session named TUCSON because the hardware updates have been completed:

```
XSTART TUCSON SESSIONTYPE(XRC) ERRORLEVEL(SESSION)
```

The SDM automatically sets the timeout value in the storage controls. For SSIDs 10FC and 10E2, it is set to TIMEOUT(00.01.00), the value specified by the last XSET command. The SDM sets all other storage controls to TIMEOUT(00.05.00).

4. Issue the following command so that XRC adds all volumes back to the session and resynchronizes them using the hardware or software bitmaps:

```
XADDPAIR TUCSON SUSPENDED
```

Identifying XRC system interactions

The following topics address operational considerations when extended remote copy operations interact with other MVS system components.

XRC varies volumes offline and online

As part of normal operations, XRC ensures that the changes on these secondary volumes are identified to the SDM system. To accomplish this, XRC issues commands on the SDM system to vary secondary volumes offline and online. These actions occur at the following times:

- During initial volume copy, when XRC copies cylinder 0, track 0 to the target volume, the secondary volume is varied offline.
- After a volume pair becomes duplex, XRC varies the secondary volume online.
- During the XRECOVER function, when the secondary volume's serial number is clipped to the primary volume serial number, the secondary volume is varied online to ensure that the volume's in-storage copy of its VTOC and VTOC index is updated.

Other systems that share these secondary volumes will only have their in-storage VTOC pointers updated when these commands are issued.

If I/O errors occur when accessing the VTOC from an attached system, vary the secondary volumes offline and online to correct the condition.

Using the DEVSERV and IDCAMS functions

Determine the status of the storage control's dual copy pair, cache, and NVS by using the DEVSERV and IDCAMS functions. These functions do not report XRC copy volume pairs or states. Issuing the XQUERY TSO command from the SDM system is the only way to obtain information about XRC volumes.

Chapter 8. Managing coupled extended remote copy sessions

An enhancement to XRC called Coupled Extended Remote Copy (CXRC) allows XRC sessions to be coupled together to guarantee that all volumes are consistent to the same time across all coupled XRC sessions. With CXRC, the size of configurations XRC can handle increases. CXRC can support installations that have thousands of volumes, whereas XRC is designed to support installations that have hundreds of volumes. An enhancement to CXRC also provides the function of being able to cluster XRC coupled sessions within a single logical partition, thus allowing more XRC sessions to be coupled to a master session.

In this topic

This topic provides information about CXRC and includes the following sections:

Overview of CXRC

Coupled Extended Remote Copy (CXRC) is an enhancement to XRC that provides the following benefits:

- Supports large environments that have an expanded number of primary storage controls and DASD volumes, in excess of those supported by a single system data mover (SDM) configuration.

  Note: A large environment can be one that has more than 20 primary storage controls or more than 1000 primary volumes, or both. Installations may have configurations that consist of thousands of volumes in multiple XRC sessions.
- Provides consistent coordination between XRC sessions to help ensure that all of the volumes can be recovered to a consistent point in time.
- Minimally impacts performance to application programs and facilitates rapid recovery in the event of a disaster.
- Allows a master session to consist of any combination of up to 14 individual XRC sessions and cluster sessions. A cluster session can contain up to 13 individual XRC sessions, allowing for a maximum of 182 (14 times 13) individual XRC sessions to be coupled to the same master session.

If there are sufficient processor and memory resources to handle peak workload, you can run up to 20 XRC sessions in a single LPAR. The sessions may or may not be coupled. Up to 13 XRC sessions on an LPAR can be combined as a single cluster session to be managed by a single CXRC master session if consistency across sessions is required.

Some XRC commands operate only on the coupled group of sessions instead of affecting individual sessions.

For additional information about CXRC TSO commands, refer to Chapter 5, "Extended remote copy command descriptions," on page 69.

Configuring a coupled XRC environment

This section includes information about the following topics:

- The types of sessions configured in a coupled XRC environment
- The CXRC configurations allowed with the SDM installed at the recovery site or at the primary site
- The considerations for defining data sets in a coupled XRC environment

Coupled Extended Remote Copy (CXRC) support is also referred to as coupled system data movers (SDMs). The coupled support allows multiple XRC sessions to be combined into a single logical session. With CXRC, multiple SDMs coordinate their consistency group processing such that the recovery of all secondary volumes can be done to a single point of consistency. A system data mover will not write data to secondary devices that have a timestamp that is past the time that all SDMs have written to journal data sets.

With CXRC, an XRC environment can be scaled to support configurations significantly greater than that possible with a single SDM. You can couple a maximum of 14 SDMs. If clustering is enabled through the PARMLIB parameter of ClusterMSession, and there are sufficient processor and memory resources to handle peak workload, CXRC can manage up to 182 SDMs, where a single OS/390 or z/OS image (a logical partition when in LPAR mode) can have a single cluster session managing up to 13 SDMs. Each cluster session acts as a single SDM to CXRC.

In defining a CXRC configuration, each SDM is set up as if it is a stand-alone independent SDM. To couple the XRC sessions, issue the XCOUPLE command to each of session.If clustering is enabled, the system is able to couple the individual XRC session specified on the XCOUPLE command to the master session through the cluster session within an LPAR. A cluster session represents a single SDM in a master session.In addition, a CXRC configuration requires a master data set, which is accessible by the SDMs. If you want to couple XRC sessions through a cluster session, a CXRC configuration requires a cluster data set.

Identifying sessions in a coupled environment

Table 31 describes session types that are used in a coupled environment.

Table 31. Coupled session descriptions

Session	Description
Active session	An XRC session in which the system data mover is running. Volumes can be added or deleted. You can issue the XADVANCE command and the XCOUPLE command with ADD and DELETE parameters to individual active XRC sessions.
Coupled session	An active individual XRC session has been added to a master session.

Table 31. Coupled session descriptions (continued)

Session	Description
Inactive session	A session in which the system data mover is not active. You can issue the XADVANCE, XSTART, XRECOVER, and the XCOUPLE commands with the PURGE parameter to individual inactive XRC sessions.
Interlocked session	A coupled XRC session in a master session that has a consistency time before the master journal time and a journal time equal to or after the master recoverable time. The master recoverable time is the minimum of the journal times reported by all XRC sessions. This is the latest time to which all sessions could recover their data in the event of a disaster, and still provide coupled consistency. XRC will recover all interlocked sessions such that all secondary volumes across the coupled interlocked sessions will be consistent. Because coupled consistency requires all volumes to be at the same time, no session is allowed to apply data which is after the master recoverable time.
Master session	A logical entity that is used to coordinate session commands and data consistency across multiple XRC sessions. A master session exists as long as there is an XRC session coupled to the master session. You can issue the XQUERY, XEND, XSUSPEND, and XCOUPLE RELEASE commands to a master session.
Cluster session	A logical entity that manages a group of coupled sessions within a single logical partition, and represents this group as a single coupled session in a master session (for example: single SDM). XRC commands cannot be issued to an active (enabled) cluster session.
Suspended session	You can issue the XADVANCE command, XSTART command, XRECOVER command, and the XCOUPLE command with the PURGE parameter to individual suspended XRC sessions.
XRC session	A logical entity that is created for processing XRC commands.

For additional information about XRC sessions as they apply in a uncoupled environment, refer to “XRC session state descriptions” on page 30.

Choosing a coupled XRC configuration

With CXRC, you can choose to use the SDM to manage your coupled data sets at the recovery site, at the primary site, or on a remote host apart from the application host.

Figure 6 on page 210 shows a coupled XRC configuration with the system data mover at the recovery location. Figure 7 on page 211 shows a coupled XRC configuration with the system data mover at the primary site. Figure 8 on page 212 shows a coupled XRC configuration with the system data mover located at the primary site but on a remote host apart from the primary host system itself.

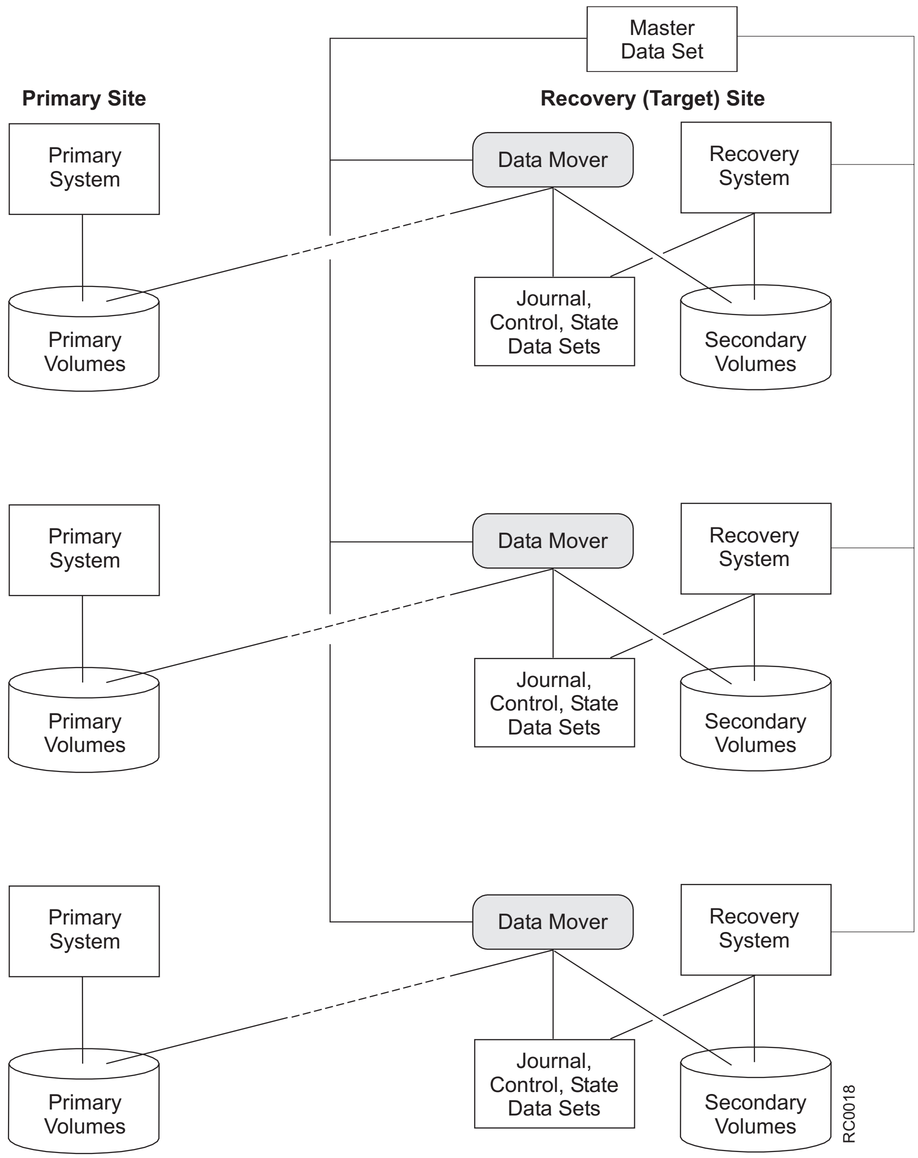


Figure 6. CXRC configuration with system data mover at recovery site

Locating the system data mover at the recovery site is the most viable scenario for effective disaster recovery, as shown in Figure 6. Because the system data mover is not at the primary site, it is protected from any disasters that might negatively impact the primary site, such as floods, storms, fires, earthquakes, bombs, or other accidental or intentional catastrophes. To determine the location of the recovery site, consider the types of disasters that you want to protect your installation's data against. For example, if hurricanes are common in your area, you might want to establish the recovery site at a sufficiently remote location, perhaps 50–100 miles away from the primary site. The flooding and power outages that accompany a hurricane, which might adversely affect your primary site, would be unlikely to affect your remote recovery site.

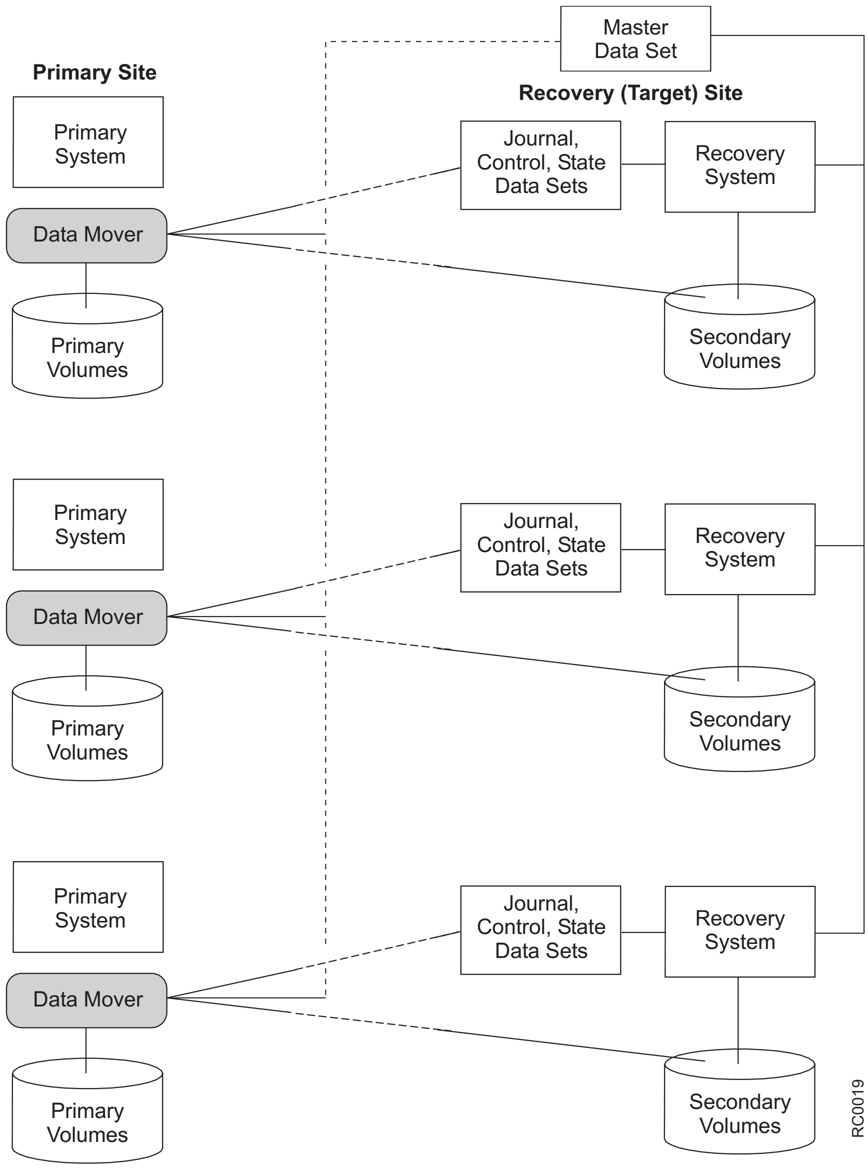


Figure 7. CXRC configuration with system data mover at primary site

You may choose to locate the system data mover on the primary host system at the primary site, with the recovery host system located remotely from the primary system, as depicted in Figure 7. However, this is the least effective and most vulnerable scenario for disaster recovery. If your application host goes down, the system data mover will be unavailable for copy operations to the recovery site. This scenario might be acceptable for short-term testing purposes only.

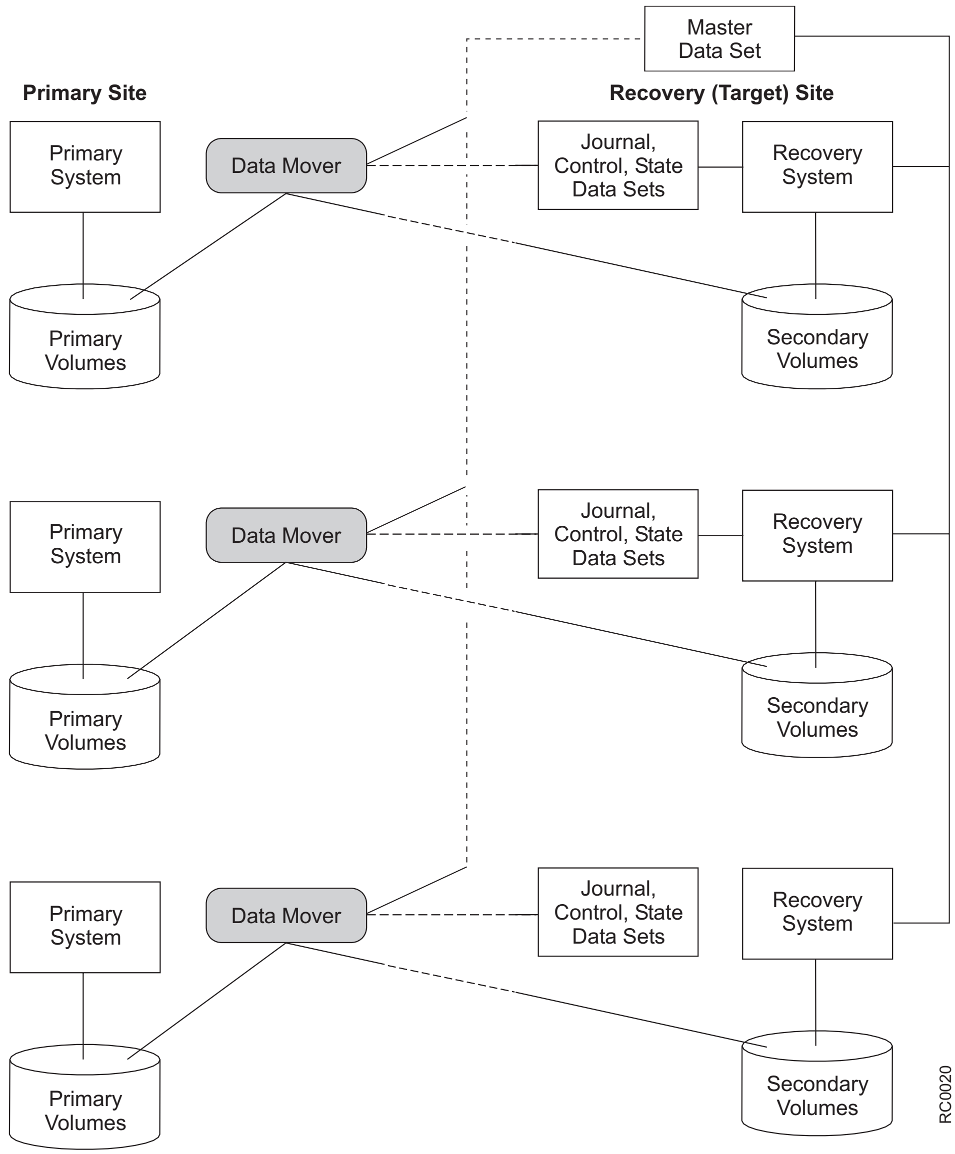


Figure 8. CXRC configuration with system data mover at primary site on remote host

Locating the system data mover on a remote, separate host at the primary site is an option for configuring your CXRC environment, as shown in Figure 8. There is a certain measure of safety built into having your SDM located on a separate host from your primary system, even if the hosts are at the same site. You run the risk, however, of the same disasters impacting the remote recovery host that affect the primary application host. This option is valid for early CXRC configuration testing or for special testing at any time.

System data mover boundaries

Ideally, it is best to split volumes so that the SDMs receive an equal workload distribution, either by the disk subsystem or by the subsystem identifiers (SSIDs). In general, most customers split mirrored volumes among multiple SDMs at the disk subsystem level to simplify the allocation of volume pairs to an SDM. If you prefer a finer level of granularity, then splitting at the SSID level is also feasible. Assigning volumes to different SDMs on a logical storage subsystem (LSS) boundary should give you enough options to reasonably distribute activity over available SDMs.

Accessing the CXRC master data set

In a CXRC environment, the SDMs communicate with each other using the master data set. At regular intervals, the SDM writes data to the master data set. This data includes the last consistency group written to the secondary volumes and the last consistency group written to the journal data sets. In the same I/O operation, the SDM reads the status of the last consistency group written to the secondary volumes and the last consistency group written to the journal data set by all other SDMs in the coupled sessions. Figure 9 illustrates the master data set concept.

There is a value in the tuning table that controls the frequency of I/Os to the master data set. If the value is set too high, delays will result that slowdown coupled SDM processing.

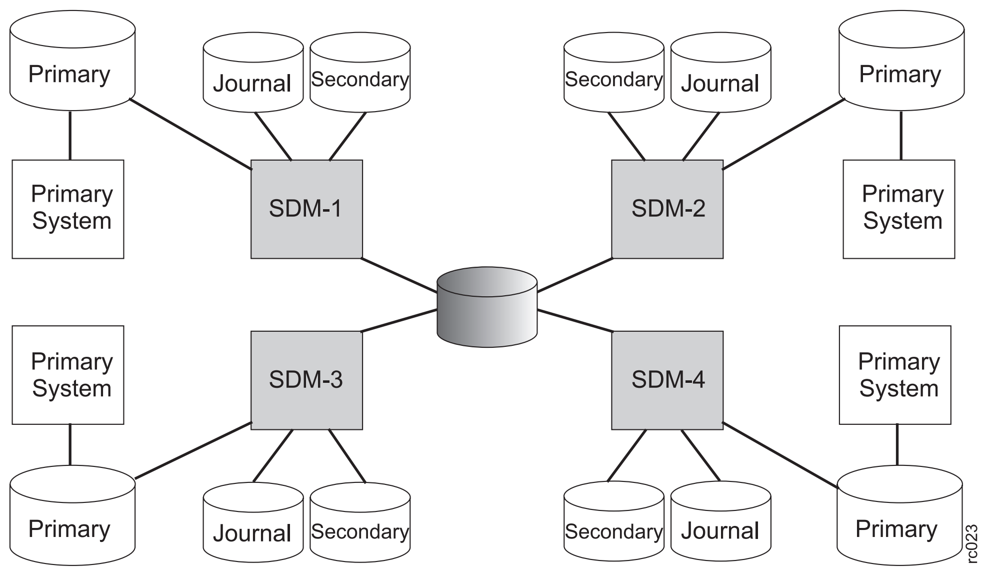


Figure 9. Master data set

Migrating CXRC session clustering

To migrate the existing CXRC sessions from individual coupling to clustering, do the following steps:

1. Suspend the master session by using the TIMEOUT parameter and install the program temporary fix (PTF).
2. Issue the XCOUPLE PURGE command to clear the information about the individual sessions from the master session.
3. Modify the ANTXIN00 parmlib member to associate each data mover LPAR with a master session and a unique cluster name. To modify the ANTXIN00 parmlib member, use the ClusterMSession and the ClusterName parameters in the STARTUP category. Here is an example:

```
STARTUP -
ClusterMSession(TUCSON) -
Clustername(SYSTEM1 CLUSTER1 - SYSTEM2 CLUSTER2 - SYSTEM3 CLUSTER3)
```

4. Modify the parmlib to enable the cluster monitoring to data set (optional).
5. Allocate and initialize a cluster control data set for each LPAR that operates as a cluster. Use the following JCL as an example:

```
//STEP1 EXEC PGM=IEBDG
//SYSPRINT DD SYSOUT=*
//DD1 DD DSN=mhlq.XCOPY.clustername.CLUSTER,
// DCB=(RECFM=FB,LRECL=15360,BLKSIZE=15360,DSORG=PS)
// SPACE=(CYL,(1,0)),STORCLAS=XRCSTATE,DISP=(NEW,CATLG)
//SYSIN DD *
DSD OUTPUT=(DD1)
CREATE QUANTITY=30
END
```

Note: You cannot use the extended format data set. You can place the data set on the same volume as any state data set accessible by the cluster LPAR.

6. To collect the cluster-level monitor data, allocate a state data set for each cluster. Use the following JCL as an example:

```
//STEP2 EXEC PGM=IEFBR14
//DD1 DD DSN=mhlq.XCOPY.clustername.STATE,
// DCB=(RECFM=F,LRECL=4096,BLKSIZE=4096,DSORG=PO),
// DSNTYPE=LIBRARY,
// SPACE=(CYL,(2,0,1)),STORCLAS=XRCSTATE,DISP=(NEW,CATLG)
```

   The data set must be a PDSE data set. The data set can be placed on the same volume as any state data set accessible by the cluster LPAR. In the XRC parmlib, ensure that the MONITOR category for each individual XRC session in the cluster has MonitorOutput(ON). If a MonitorWakeup value is specified, use the same value for all individual XRC sessions in the cluster.

7. You can attain the following results by changing the XRC parmlib to use a smaller SHADOW READDELAY value, such as 500:
   - Achieve more aggressive Recovery Point Objective targets.
   - Avoid delays from lightly loaded data movers in large scale configurations.

   Note: This operation results in higher CPU utilization during periods of non-peak workloads.

8. Issue the F ANTAS000,CREFRESH command to each LPAR to initiate cluster setup processing.
9. Restart individual sessions and resynchronize.
10. When duplex is reached, issue the XCOUPLE ADD command to couple individual sessions to the master session.

    Note: Do not use the cluster name in the XCOUPLE commands.

Note: Before increasing the number of XRC data movers per LPAR, ensure that sufficient system resources (processor and memory) are available to handle the expected simultaneous peak data mover workload. Workload demands and system constraints might significantly reduce the number of data movers that can run effectively on a given LPAR.

To migrate the existing CXRC sessions from clustering to individual coupling, do the following steps:

1. Suspend master session by using TIMEOUT parameter.
2. Issue XCOUPLE PURGE command to clear information about all individual XRC sessions that are previously coupled through the cluster.
3. Modify the ANTXIN00 parmlib member to specify ClusterMSession (DISABLED).

4. Issue F ANTAS000,CREFRESH command to restart the cluster setup processing and disable clustering.
5. Issue XCOUPLE ADD command to couple the individual XRC sessions to the master session.

Note: You can only migrate the existing CXRC sessions from clustering to individual coupling when the coupled session has fewer than 14 individual XRC session members.

Defining master data sets in a coupled environment

Before activating XRC in a coupled environment, you need to allocate the master data set. This section provides instructions for changing the characteristics of master data sets after they have been established.

Allocating the CXRC master data set

All of the XRC sessions coupled in the master session continuously write to and read from the master data set. This access allows communication between hosts when data is copied from the primary site to the recovery site.

When evaluating the resources you need to successfully configure a CXRC environment, you must plan so that all primary hosts and recovery hosts have access to the master data set.

Rules: You must follow these conditions in order to create the necessary access:

- Catalog the master data set and make it accessible to each host system that processes a coupled session, as well as to the system that processes the XRECOVER command. The master data set name is *mhlq.XCOPY.msession.MASTER*.
- Ensure that the master data set does not reside on the same volume as the control data set and the volume that contains the journal data sets.
- Allocate the master data set without defining secondary extents.
- Allocate the master data set as physical sequential and not striped.
- Allocate the master data set with contiguous tracks as the I/O is assumed not to span multiple extents. Access to the master data set is controlled using GRS enqueues, and therefore, all systems with coupled SDMs must be in the same GRS configuration.
- Allocate the master data set on a single disk device.
- Ensure that the required size of the master data set is one cylinder, which allows 14 XRC sessions to be coupled. The required size of the master data set is fixed, which also fixes the maximum number of sessions that can be coupled into one master session.
- Preallocate the master data set size before use of the XCOUPLE command. Only the space that is allocated at the time the XCOUPLE ADD command is issued is available for XRC use.
- Protect the master data set from unauthorized use.
- Do not allocate the master data set in the Extended Address Space (EAS) of an Extended Address Volume (EAV).

Allocate the master data set with one cylinder primary space and zero cylinders secondary space as follows:

```
DCB=(RECFM=FB,LRECL=15360,BLKSIZE=15360,DSORG=PS),SPACE=(CYL,(1,0))
```

For additional information about protecting the master data set using RACF, refer to "Controlling access to XRC commands" on page 51.

For additional information about configuring the master data set, refer to "Specifying the master data set."

Changing the characteristics of a master data set

XRC cannot dynamically accept changes to the master data set. To change the characteristics of the master data set when it has been established, follow this procedure:

1. Remove all coupled sessions from the master session (XCOUPLE DELETE command for active sessions or XCOUPLE PURGE command for inactive sessions).
2. Reallocate the master data set.
3. Add sessions back to the master session (XCOUPLE ADD command).

Specifying the master data set

XRC uses the master data set to keep track of multiple XRC sessions that are coupled to the same master session that is associated with the master data set. The master data set must be accessible from, and cataloged on, all SDM hosts and all recovery hosts.

XRC uses the master data set name in one of the following forms:

- `SYS1.XCOPY.`*`msession_id`*`.MASTER`
- *`mhlq`*`.XCOPY.`*`msession_id`*`.MASTER`

Where:

msession_id
: Specifies the same name that you issued with the MSESSION parameter of the XCOUPLE ADD command. The name can be up to eight characters long. The name you specify for the *msession_id* cannot be the same as any *session_id*.

mhlq
: Specifies the name of the master data set high-level qualifier that is specified on the MHLQ parameter of the XCOUPLE ADD command. The *mhlq* must be one to eight characters, and can be any name acceptable to TSO. "SYS1" is the default value.

  You can change the default *mhlq* to a name that corresponds with the MHLQ that is issued on the XCOUPLE command. When you issue the XSUSPEND, XEND, or XQUERY commands with the MASTER option, use the MHLQ parameter. You can also use the MHLQ parameter on the XRECOVER and XQUERY commands to override the defaults.

The following is an example of a master data set name:

```
SYS1.XCOPY.DALLAS.MASTER
```

Defining cluster data sets in a coupled environment

Before enabling XRC to use cluster sessions in a coupled environment, you need to allocate the cluster data set. This section indicates when a cluster data set is needed, and how to allocate a cluster data set.

When cluster data sets are needed

A cluster data set is required if multiple SDMs within a logical partition are to be coupled to a master session through a cluster session. The XRC PARMLIB parameter of ClusterMSession of the STARTUP category is used to activate (enable) a cluster session in a logical partition. The value provided for this parameter indicates the master session name to which SDMs in the logical partition are to be coupled. All SDMs logical partition for which an XCOUPLE add command is issued are to be coupled to the specified master session through the cluster session.

The cluster session and not the individual SDMs in a logical partition continuously write to and read from the master data set. The cluster data set contains information that allows a cluster session to restart if the ANTCL*nnn* address space terminates (the address space in which the cluster session runs). This information includes the name of the master session with which the cluster session was last associated.

Allocating the cluster data set

The cluster data set must be allocated if a cluster session is to be used in an LPAR.

Rules: You must follow these conditions in order to create the necessary access:

- Catalog the cluster data set and make it accessible to the host system (LPAR) that is to have a cluster session, as well as to the system that processes the XRECOVER command for each of the SDMs that will be coupled throughthe cluster session. The cluster data set name is *mhlq.XCOPY.ClusterName.CLUSTER*. The *mhlq* is the same high level qualifier as used for the master data set. The *ClusterName* is the value specified on XRC PARMLIB parameter of ClusterName of the STARTUP category, or is defaulted to the host system name.
- Ensure that the cluster data set can reside on the same volume as the master data set.
- Allocate the cluster data set without defining secondary extents.
- Allocate the cluster data set as physical sequential and not striped.
- Allocate the cluster data set with contiguous tracks as the I/O is assumed not to span multiple extents. Access to the cluster data set is controlled using GRS enqueues, and therefore, all systems with coupled SDMs must be in the same GRS configuration.
- Allocate the cluster data set on a single disk device.
- Ensure that the required size of the cluster data set is one cylinder. The required size of the cluster data set is fixed.
- Preallocate the cluster data set before specifying master session name on the ClusterMSession PARMLIB parameter. Only the space that is allocated at the time the ANTCL*nnn* address space is started will be available for XRC use.
- Protect the cluster data set from unauthorized use.
- Do not allocate the cluster data set in the Extended Address Space (EAS) of an Extended Address Volume (EAV).

Allocate the cluster data set with one cylinder primary space and zero cylinders secondary space as follows:

```
DCB=(RECFM=FB,LRECL=15360,BLKSIZE=15360,DSORG=PS),SPACE=(CYL,(1,0))
```

Changing the characteristics of a cluster data set

XRC cannot dynamically accept changes to the cluster data set. To change the characteristics of the cluster data set, follow this procedure:

1. Remove all coupled sessions from the cluster session (XCOUPLE DELETE command for active sessions or XCOUPLE PURGE command for inactive sessions). Issue the XSTATUS command to determine the SDMs coupled through the cluster session.
2. Reallocate the cluster data set.
3. Issue the XCOUPLE ADD command for each of the sessions previously coupled through the cluster session.

Specifying the cluster data set

XRC uses the cluster data set to keep track of master session to which all SDMs in a cluster session are coupled. The cluster data set must be accessible from, and cataloged on, the host on which the cluster session is to be active and all recovery hosts for all SDMs in the cluster session.

XRC uses the cluster data set name in the following form:

- `mhlq.XCOPY.ClusterName.CLUSTER`

Where:

ClusterName
: Specifies the value specified on XRC PARMLIB parameter of ClusterName of the STARTUP category, or defaults to the host system name. The name can be up to eight characters long. The name you specify for the *ClusterName* cannot be the same as any XRC *session_id*.

mhlq
: Specifies the master data set high-level qualifier that is used to access the master data set. The *mhlq* must be one to eight characters, and can be any name acceptable to TSO.

The following name is an example of a cluster data set name:

```
SYS1.XCOPY.SYSTEM1.CLUSTER
```

Using journal, state, and control data sets

There are special considerations for journal, state, and control data sets when you work with master data sets in CXRC. In a CXRC environment, each SDM uses its own journal, state, and control data sets. Place these data sets so a performance bottleneck is not created due to contention between the multiple SDMs for the same channel path, disk subsystem, or volume resources.

Example: Suppose you have four primary disk subsystems, generating 100 MB per second update rate and two separate SDMs, each managing two disk subsystems. By placing the journal data sets for both SDMs on a common disk subsystem, that subsystem will experience a 400 MB per second update rate. The channel paths, disk subsystem, and volumes must be able to manage this update rate.

Guidelines: Consider the following guidelines when using journal, state, and control data sets in a coupled environment:

- Allocate the journal, control, and state data sets for the individual sessions in the same manner as the existing support.
- Add a new member to the state data set to indicate that the session is coupled and to allow the XSTART command to process the XCOUPLE ADD command automatically during restart. On a new start, this member is used to uncouple the session.

Note: The size of this new state data set member is small and does not affect the size requirements for the state data set.

- Define from eight to 16 journal data sets to improve the efficiency of XRC in a large coupled environment.
- Allocate the control data set using physical sequential allocation.

  Note: The state data set is updated whenever you issue XCOUPLE ADD, XCOUPLE DELETE, and XCOUPLE PURGE commands.

If there is insufficient space in the state data set, you will not be able to couple an XRC session to a master session. To allocate a new, larger state data set, perform the following steps:

1. Issue XSUSPEND *msession_id* TIMEOUT(*hh.mm.ss*).
2. Copy all members from the existing state data set to the new, larger data set.
3. Rename the existing state data set to an unused name.
4. Rename the new, larger data set to the name of the pre-existing state data set.
5. Issue the XSTART command and resume XRC operations.

Managing coupled XRC operations

This section contains the following information about managing coupled XRC operations:

With CXRC session clustering, you can enable a master session to consist of any combination of up to 14 individual sessions and clusters. This function enables a maximum of 182 (14 times 13) individual XRC sessions to be coupled to a single CXRC master session.

Before you begin: You need to be familiar with the commands that are used in the scenarios in this section. For more information, refer to Chapter 5, "Extended remote copy command descriptions," on page 69.

Coupling XRC sessions

Use the XCOUPLE ADD command to couple an XRC session to a master session. The first time you issue this command for a session, XRC creates a new member in the session's state data set to indicate that the session is coupled to the specified master session. The member is also used during XSTART processing to determine if the session is to be recoupled to a master session. The XCOUPLE ADD command may only be invoked against XRC sessions that have been started with the SESSIONTYPE(XRC) parameter.

You can couple multiple SDMs within a logical partition to the same master session through a cluster session. Specifying the master name on the XRC PARMLIB parameter of ClusterMSession enables a cluster session for a logical partition, and all XCOUPLE ADD commands issued on this logical partition cause the SDMs to be coupled to the same master session through the cluster session.

Note: You should not start multiple SDMs in the same logical partition unless you have sufficient real storage to support them. A busy SDM can effectively consume all usable storage in a 2 GB storage configuration. Starting multiple SDMs and coupling them together in this environment can cause a negative performance impact.

Recommendation: The best performance can be achieved by ensuring that all volumes in a data mover session are in DUPLEX state prior to coupling the data mover session to a Master session. The time for the volumes to reach DUPLEX state in an uncoupled data mover can be up to 50% faster than the time for volumes in a coupled data mover to reach DUPLEX state.

Example: coupling new or existing XRC sessions

The following scenario describes the XRC commands that are used to start a new master session by coupling two XRC sessions. One of the XRC sessions is an existing session and the other is a new session. You can issue commands to the same logical partition or to different logical partitions. The master data set for master session CXRC must be accessible from each logical partition.

Perform the following steps to couple new or existing XRC sessions:

1. Start a session named TUCSON, setting the error level for the session using the ERRORLEVEL parameter by entering:

   ```
   XSTART TUCSON SESSIONTYPE(XRC) ERRORLEVEL(SESSION)
   ```

   This ensures a recoverable consistency across all coupled sessions. See “Specifying the ERRORLEVEL parameter in coupled and uncoupled sessions” on page 255 for information about what happens when an error occurs during master session processing for a coupled session.
2. Issue the following command to add the XRC session named TUCSON to the master session named CXRC, with a default master high-level qualifier of SYS1.

   ```
   XCOUPLE TUCSON ADD MSESSION(CXRC) SYS1
   ```

3. Start a previously suspended session named PHOENIX.

   ```
   XSTART PHOENIX SESSIONTYPE(XRC) ERRORLEVEL(SESSION)
   ```

4. Add the XRC session named PHOENIX to the master session named CXRC, with a default master high-level qualifier of SYS1.

```
XCOUPLE PHOENIX ADD MSESSION(CXRC)SYS1
```

5. Add an XRC utility device and two volume pairs to session TUCSON.

```
XADDPAIR TUCSON VOLUME(UTIL01,XRCUTL,PRIM01,SEC001,PRIM02,SEC002)
```

6. Add the previously suspended volumes again to the session PHOENIX.

```
XADDPAIR PHOENIX SUSPENDED
```

7. Issue the following command to generate an XQUERY MASTER report:

```
XQUERY CXRC MASTER
```

Result: The XQUERY MASTER report command generates the following output:

```
ANTQ8300I XQUERY STARTED FOR MSESSION(CXRC) MHLQ(SYS1)
ANTQ8202I XQUERY MASTER REPORT - 002
ANTQ8302I SESSION   STA VOL INT CMD   JOURNAL DELTA      RCV/ADV DELTA
ANTQ8303I -------------------------------------------------------------
ANTQ8304I TUCSON    ACT      Y       =00:00:01.830727  =00:00:00.000000
ANTQ8304I PHOENIX   ACT      Y       +00:00:00.000000  +00:00:02.352970
ANTQ8305I TOTAL=2  ACT=2  SUS=0  END=0  ARV=0  RCV=0  UNK=0
ANTQ8308I MSESSION RECOVERABLE TIME(2004.080 22:33:20.664559)
ANTQ8309I INTERLOCKED=2  NON-INTERLOCKED=0
ANTQ8301I XQUERY MASTER REPORT COMPLETE FOR MSESSION(CXRC)
```

Coupling XRC sessions through a cluster session

You can couple XRC sessions running in a logical partition to the same master session as a single entity. This single entity, or cluster session, gathers all coupled information for all XRC coupled sessions within a logical partition, and presents this information in the master data set as a single coupled session. The XQUERY command with the MASTER parameter indicates that the session is a cluster session.

You use the XRC PARMLIB parameter of ClusterMSession of the STARTUP category to indicate the master session name to which all XRC coupled sessions in a logical partition will be coupled. When a master session name is found in this parmlib parameter, every time the ANTAS000 address space is started or restarted, the ANTCL*nnn* address space is started to manage a cluster session for a logical partition. Then, each XRC session that is specified on an XCOUPLE command issued with the ADD parameter the system couples to this master session through this cluster session. All XRC coupled sessions in logical partition must be coupled through the cluster session.

The XRC PARMLIB parameter of ClusterName of the STARTUP category can be used to specify the name of the cluster session. If you do not specify this parameter, the default name is the system host name. The cluster session name is used to determine the name of the cluster data set that is required for a cluster session.

The cluster data set is used to determine the previous cluster session environment for a logical partition when the ANTCL*nnn* address space is being started or restarted. The information found in the cluster data set determines the behavior of a cluster session.

The following are rules guiding the use of a cluster session.
- You can couple up to 13 XRC sessions in a logical partition through a cluster session. The first XRC session coupled through a cluster session causes the cluster session to be coupled to the master session. When the last session in a cluster session is uncoupled, the cluster session will be uncoupled from the master session.
- You cannot issue XRC commands specifying an active cluster session name for the *session_id*.
- The cluster session reflects to the master session the combined status of all XRC sessions coupled through the cluster session.
- The system does not start a cluster session during startup of the ANTAS000 address space or through the MVS console command of **F ANTAS000,CREFRESH** if one of the following conditions exists:
  - The previous master session used by the cluster is not the same as the master session specified in the ClusterMSession (including the name and hlq). XRC suspended or active sessions are currently coupled through this cluster session. XRC uses the cluster data set to make this determination.
  - The cluster data set or the master data set cannot be located through normal catalog search, or either of these data sets cannot be read. The XRC PARMLIB parameter ClusterName might have been changed, or the cluster name specified on the CREFRESH command may be invalid.
  - The cluster session is currently active on another logical partition. The SYSXRCLS enqueue for the mhlq.clustername was enqueued by another system in the complex.
  - Storage cannot be obtained for the ANTCL*nnn* address space, or the address space failed to be created.
- You can also issue XRC commands allowed for coupled sessions that are not in a cluster session to the XRC coupled sessions in a cluster session.
- A cluster session will not be disabled during startup of the ANTAS000 address space or through the MVS console command of 'F ANTAS000,CREFRESH' if one of the following conditions exists (the cluster session will be enabled using the previous settings for the cluster session):
  - The cluster session was previously enabled and one or more XRC sessions were coupled through the cluster session to a master session.
  - The XRC PARMLIB parameter of ClusterMSession specifies a master session name that is different then the master session name that the cluster session was coupled to when the ANTCL*nnn* address space last terminated, and one or more XRC sessions were coupled through the cluster session.
- A cluster session may be started/restarted using a different ClusterMSession value if no XRC sessions are currently coupled through the cluster session.
- An XCOUPLE command specified with the ADD parameter will fail if one of the following conditions is true:
  - The MSESSION value specified is different from the ClusterMSession value.

- 13 XRC sessions are already coupled through the cluster session.
- The master session has 14 coupled sessions.
- The ANTCL*nnn* address space is not active and the ClusterMSession indicates that a cluster session is to be used for the logical partition. This may be a transition situation or it may indicate that an error occurred attempting to create the ANTCL*nnn* address space.
- The cluster session encounters an error processing the request.

Note:

Functions described in this chapter that can be performed for coupled sessions may also be performed for XRC sessions which are coupled through a cluster session. These functions cannot be performed using the cluster session name on the commands. You must issue the requests specifying the XRC sessions names, and the result will be provided to the master session through the cluster session.

In addition, any error that occurs for one or more XRC sessions that are coupled through a cluster session will cause the cluster session to indicate to the master session that the cluster session encountered an error. If different errors occurred, the XQUERY MASTER report may indicate the status of MIX. Issue the XSTATUS CLUSTER command to determine the status of each session.

Example: coupling existing coupled XRC sessions through a cluster session

The following scenario describes the steps needed to couple existing coupled XRC sessions using a cluster session to the same master session. The scenario provides the steps for converting existing coupled sessions for a single logical partition. These same steps can be used for other logical partitions. The master data set for master session CXRC must be accessible from the logical partition. In this example, the XRC sessions are currently coupled to the CXRC master session.

Perform the following steps to couple existing XRC coupled sessions through a cluster session:

1. Allocate a cluster data set with the name **SYS1.XCOPY.ARIZONA.CLUSTER**. See"Allocating the cluster data set" on page 217 for details on how to allocate the cluster data set.
2. Update the ANTXIN00 member in parmlib to include the following XRC parmlib parameters under the STARTUP category:

```
ClusterMSession(CXRC) -
ClusterName(ARIZONA)
```

3. Issue the following XRC commands for each XRC coupled session in the logical partition. You must ensure that all coupled sessions must are first suspended before they can be coupled through a cluster session:

```
XSUSPEND session_id VOLUME(ALL)
XCOUPLE session_id DELETE
```

4. Issue the following MVS console command to activate the cluster session:

```
F ANTAS000,CREFRESH
```

5. Issue the following XRC command for each session that you previously suspended:

```
XCOUPLE session_id ADD MSESSION(msession_id)
```

Querying coupled status for XRC sessions

To determine the status of each coupled session that is associated with the specified master session, issue the XQUERY command with the MASTER parameter. The XQUERY output indicates the status of the master session, the status of each session, and the overall volume status for each session. In the case of a cluster session, the XQUERY MASTER output indicates the accumulated status of the coupled SDMs in the cluster session. Issue the XSTATUS command with the CLUSTER parameter to determine the individual status of each SDM in a cluster session. In addition, the XQUERY command that is specified with the VOLUME or SUMMARY parameter provides the master session status.

For additional information about the reports that are generated when you issue the XQUERY MASTER, VOLUME and SUMMARY commands, refer to “Examples of XQUERY reports in a coupled environment” on page 189. For examples of the XSTATUS CLUSTER command, refer to “XSTATUS – querying XRC status” on page 112.

Creating a backup copy of secondary volumes (XADVANCE)

The XADVANCE command applies updates to the secondary volumes of an XRC session to a time that is consistent with the volumes of other sessions that are coupled to the master session. This is the same time to which an XRECOVER command updates secondary volumes of a session, except that the XADVANCE command does not cause the secondary volumes to be named with (clipped to) the primary volume volser. The XADVANCE command is used to advance all secondary volumes for all coupled sessions to a consistent time. When the XADVANCE function has completed, a copy of the secondary volumes can be taken with data that is consistent to this time.

Using the XADVANCE command

Follow the directions in Table 32 for using the XADVANCE command depending on the status of the session.

Table 32. Using the XADVANCE command

If the session is . . .	Then . . .	Action . . .
Active	You can issue the XADVANCE command.	Suspend all volumes in the session and issue the command to the logical partition that the session is active in.
Inactive	You can issue the XADVANCE command.	You can issue the XADVANCE command from any logical partitions.

Table 32. Using the XADVANCE command (continued)

If the session is . . .	Then . . .	Action . . .
In COUPLE_FAILED status	The XADVANCE command will not process successfully against a session that is in COUPLE_FAILED status.	You must fix the error that occurred before you can issue the command. See "What to do if a session is in COUPLE_FAILED status" on page 234 for instructions on how to remove the COUPLE_FAILED status for a session.

Restarting an inactive XRC coupled session

You can use the following procedure to restart inactive XRC sessions. Issue the XQUERY MASTER command to determine if an XRC session is inactive. The session status of **END**, **UNK**, or **SUS** on the ANTQ8304I message indicates that a coupled XRC session is inactive. You can issue the XQUERY SUMMARY command on the system from which the XRC session is running to determine if the session is active.

To restart inactive XRC sessions, perform the following tasks:

1. Issue the XADVANCE command for all sessions that indicate that the RCV/ADV DELTA time is positive (that is, the session consistency time is consistent with the master journal time). You must issue the command on the system on which the session was running.

   Note: After the XADVANCE command successfully completes, the data on the secondary devices is consistent with the master journal time. You can now create backup copies of your secondary devices as indicated by your installation backup procedures.
2. Issue the XSTART command for each session.
3. Issue the XQUERY MASTER command (and the XSTATUS CLUSTER command if sessions are coupled through a cluster session).

   Note: You must perform steps 1 through 3 for each inactive session before issuing the XRELEASE command.
4. Issue the XCOUPLE command with the RELEASE option if the ANTQ8306I message indicates that the master session is in HOLD status. (See "What to do if a session is in COUPLE_FAILED status" on page 234 for additional information regarding the HOLD status for a master session.)
5. Issue the XADDPAIR command for each pair in the session. If volume pairs are in the SUS state (suspended), you can use the SUSPENDED option of this command to add all of these volume pairs again.

Example: restarting an inactive XRC coupled session after an error occurs

The following scenario describes how to restart an inactive coupled XRC session after an error has occurred on the session. Issue the XQUERY MASTER command to determine if an XRC session is inactive. The session status of UNK may indicate that a coupled XRC session is inactive. You can issue an XQUERY SUMMARY command on the system from which the XRC session is running to determine if the session is still active.

First, an error occurs on session PHOENIX; session TUCSON reacts and suspends all volumes to the msession recovered time.

Result: This is an example of the message that was generated:

```
ANTC8402W XRC SESSION(TUCSON) COUPLED TO MSESSION(CXRC) DETECTED
COUPLING_TIMEOUT CONDITION IN XRC_SESSION(PHOENIX),
ANTC8402W (CONT) SUSPENDING ALL VOLUMES

ANTV8107I SUSPEND COMPLETE FOR VOLUME PAIR(PRIM01,SEC001) FOR SESSION(TUCSON)
   AT CONSISTENCY_GROUP TIME(2004.080 ANTV8107I (CONT) 22:35:21.643364)
ANTV8107I SUSPEND COMPLETE FOR VOLUME PAIR(PRIM02,SEC002) FOR SESSION(TUCSON)
   AT CONSISTENCY_GROUP TIME(2004.080 ANTV8107I (CONT) 22:35:21.643364)
```

Perform the following steps to restart an inactive XRC coupled session after an error occurs:

1. Issue the following command to generate an XQUERY MASTER report:

```
XQUERY CXRC MASTER
```

   Result: The following output is generated from an XQUERY MASTER report for the master session named CXRC, with a default master high-level qualifier of SYS1:

```
ANTQ8300I XQUERY STARTED FOR MSESSION(CXRC) MHLQ(SYS1)
ANTQ8202I XQUERY MASTER REPORT - 002
ANTQ8302I SESSION   STA VOL INT CMD   JOURNAL DELTA      RCV/ADV DELTA
ANTQ8303I -------------------------------------------------------------
ANTQ8304I TUCSON    ACT SYM  Y       +00:00:03.824936  =00:00:00.000000
ANTQ8304I PHOENIX   UNK      Y       =00:00:00.000000  +00:00:03.470722
ANTQ8305I TOTAL=2  ACT=1  SUS=0  END=0  ARV=0  RCV=0  UNK=1
ANTQ8306I MSESSION STATUS=HOLD
ANTQ8308I MSESSION RECOVERABLE TIME(2004.080 22:35:21.643364)
ANTQ8309I INTERLOCKED=2  NON-INTERLOCKED=0
ANTQ8301I XQUERY MASTER REPORT COMPLETE FOR MSESSION(CXRC)
```

2. Determine why the XRC session PHOENIX did not respond so you can resolve the error. The XRC session PHOENIX might have ended for one of the following reasons:
   - The logical partition on which the XRC session PHOENIX was running had to be IPLed (Initial Program Load).
   - The XRC address space was canceled for this XRC session.
   - The logical partition where this XRC session was running lost connectivity to the master data set.
3. Correct the problem for XRC session PHOENIX.
4. Issue the following command to apply journal data to the XRC session PHOENIX. The XRC session PHOENIX has a positive RCV/ADV DELTA time. Issue the XADVANCE command on the system from which PHOENIX is running.

```
XADVANCE PHOENIX
```

The XADVANCE command brings the consistency time of the session up to the msession recoverable time.

Result: The processing of the XADVANCE command generates the following XQUERY output:

```
ANTQ8300I XQUERY STARTED FOR MSESSION(CXRC) MHLQ(SYS1)
ANTQ8202I XQUERY MASTER REPORT - 002
ANTQ8302I SESSION   STA VOL INT CMD   JOURNAL DELTA      RCV/ADV DELTA
ANTQ8303I -------------------------------------------------------------
ANTQ8304I TUCSON    ACT SYM  Y       +00:00:03.824936  =00:00:00.000000
ANTQ8304I PHOENIX   UNK      Y       =00:00:00.000000  +00:00:00.000000
ANTQ8305I TOTAL=2  ACT=1  SUS=0  END=0  ARV=0  RCV=0  UNK=1
ANTQ8306I MSESSION STATUS=HOLD
ANTQ8308I MSESSION RECOVERABLE TIME(2004.080 22:35:21.643364)
ANTQ8309I INTERLOCKED=2  NON-INTERLOCKED=0
ANTQ8301I XQUERY MASTER REPORT COMPLETE FOR MSESSION(CXRC)
```

5. Perform your installation backup procedures to back up the data on your secondary devices for both sessions. Both sessions are consistent to the msession recoverable time.
6. Issue the XSTART command to restart the XRC session PHOENIX.

```
XSTART PHOENIX SESSIONTYPE(XRC) ERRORLEVEL(SESSION)
```

7. Issue the following command to generate an XQUERY MASTER report:

```
XQUERY CXRC MASTER
```

Result: The following output is generated from an XQUERY MASTER report for the master session named CXRC, with a default master high-level qualifier of SYS1:

```
ANTQ8300I XQUERY STARTED FOR MSESSION(CXRC) MHLQ(SYS1)
ANTQ8202I XQUERY MASTER REPORT - 002
ANTQ8302I SESSION   STA VOL INT CMD   JOURNAL DELTA      RCV/ADV DELTA
ANTQ8303I -------------------------------------------------------------
ANTQ8304I TUCSON    ACT SYM  Y       +00:00:03.824936  =00:00:00.000000
ANTQ8304I PHOENIX   ACT ERR  Y       =00:00:00.000000  +00:00:00.000000
ANTQ8305I TOTAL=2  ACT=2  SUS=0  END=0  ARV=0  RCV=0  UNK=0
ANTQ8306I MSESSION STATUS=HOLD
ANTQ8308I MSESSION RECOVERABLE TIME(2004.080 22:35:21.643364)
ANTQ8309I INTERLOCKED=2  NON-INTERLOCKED=0
ANTQ8301I XQUERY MASTER REPORT COMPLETE FOR MSESSION(CXRC)
```

8. Issue the XCOUPLE command with the RELEASE option to release the master session named CXRC, with a default master high-level qualifier of SYS1, from HOLD status.

```
XCOUPLE CXRC RELEASE SYS1
```

See “What to do if the master session is in coupled HOLD status” on page 228 for additional information regarding the HOLD status for a master session.

9. Issue the XQUERY command after you issue the XCOUPLE command to ensure that the status of the master session named CXRC, with a default master high-level qualifier of SYS1, is no longer in HOLD status.

```
XQUERY CXRC MASTER
```

Result: The command will generate the following output:

```
ANTQ8300I XQUERY STARTED FOR MSESSION(CXRC) MHLQ(SYS1)
ANTQ8202I XQUERY MASTER REPORT - 002
ANTQ8302I SESSION   STA VOL INT CMD   JOURNAL DELTA      RCV/ADV DELTA
ANTQ8303I -------------------------------------------------------------
ANTQ8304I TUCSON    ACT AVS  N
ANTQ8304I PHOENIX   ACT AVS  N
ANTQ8305I TOTAL=2  ACT=2  SUS=0  END=0  ARV=0  RCV=0  UNK=0
ANTQ8308I MSESSION RECOVERABLE TIME(NO_TIME_AVAILABLE)
ANTQ8309I INTERLOCKED=0  NON-INTERLOCKED=2
ANTQ8301I XQUERY MASTER REPORT COMPLETE FOR MSESSION(CXRC)
```

10. Issue the following command to add the volume pairs to the XRC session PHOENIX:

```
XADDPAIR PHOENIX(VOL1,VOL2)
```

What to do if the master session is in coupled HOLD status

A CXRC master session is placed into the coupled HOLD status when XRC determines that the master recoverable time must remain unchanged to preserve the data consistency across all of the interlocked XRC sessions coupled to the master session.

XRC provides this data consistency function to allow manual backup of data for these sessions. The master recoverable time is the consistency time to which these coupled sessions can be recovered. This master recoverable time is protected and will not be changed when the master session is in the coupled HOLD status. XRC also disallows all actions that cause the master recoverable time to change.

Determining coupled HOLD status

You can issue the XQUERY command with the MASTER parameter to determine if a master session is in the coupled HOLD status.

Entering coupled HOLD status

The master session enters the coupled HOLD status when one of the events in Table 33 occurs.

Table 33. Triggering events for coupled hold status

If the event is . . .	Then . . .
XRECOVER command	For the first XRECOVER command you issue for any of the coupled sessions associated with the master session, XRC places the master session in the coupled HOLD state to ensure a consistent recovery time for all sessions. The master session remains in HOLD status until the last coupled session is recovered.

Table 33. Triggering events for coupled hold status (continued)

If the event is . . .	Then . . .
Error in Session	An interlocked coupled session that detects an error suspends all duplex volumes in the session and places the master session in HOLD status.
Abnormal End or Cancel of a Session	A coupled session that abnormally ends (ABENDs) or is ended by a cancel command can cause the master session to be placed in HOLD status when the coupled session is restarted.
Sympathetic Suspend of Session	An interlocked coupled session suspends all volumes in the session if one of the following occurs: • The interlocked coupled session detects that all volumes have been suspended for another interlocked coupled session, because of an error. • The interlocked coupled session detects that another interlocked coupled session has not updated the master session for some time. XRC changes the master session state to coupled HOLD state when this occurs. The following are situations that will cause a sympathetic suspend: • An error occurs to a volume of a duplex volume pair that was added with the ERRORLEVEL(SESSION) parameter. • An error occurs to a session that forces the session's address space to be restarted, such as a session level error. • A condition occurs where a session is not updating the master data set to indicate a current session consistency time. See "What to do if a session is in COUPLE_FAILED status" on page 234 for a description of the COUPLE_FAILED status.

Restrictions when the master session is in coupled HOLD status

The following commands are not allowed for any of the coupled sessions while the master session is in the coupled HOLD status because their functions would cause the master recoverable time to change:

- XCOUPLE with the ADD parameter for a new session
- XADDPAIR

Clearing a master session from coupled HOLD status

Before you begin: Before clearing a master session from the coupled HOLD status, you must follow the procedure described in "Restarting an inactive XRC coupled session" on page 225.

Attention: After you have issued the XCOUPLE command with either the RELEASE, DELETE, or PURGE parameter, data for the specified session may not be consistent with the other sessions in the master session.

Clear the coupled HOLD status of a master session with one of the methods listed in Table 34 on page 230.

Table 34. Clearing coupled HOLD status

When . . .	Then . . .	Notes . . .
All XRECOVER commands complete processing	The HOLD status clears automatically.	none
An error was detected for a session	After you have corrected the error and performed the necessary backup procedures, issue the XCOUPLE RELEASE command.	See "Using the XCOUPLE RELEASE command" on page 233.
An error was detected for a session, and the error cannot be fixed or determined	Uncouple all sessions by issuing the XCOUPLE DELETE command for active sessions and the XCOUPLE PURGE for all inactive sessions.	See "Using the XCOUPLE DELETE and XCOUPLE PURGE commands" on page 235.

Example: resuming operations after a sympathetic suspension

An error has occurred in the coupled session named TUCSON. Because you specified the error level as SESSION on the ERRORLEVEL parameter when you issued the XSTART command, XRC suspends all volume pairs in TUCSON. The session named PHOENIX issues an ANTC8402W message to indicate that a sympathetic suspend must be done and why. The session named PHOENIX suspends all volumes in the session.

The following scenario describes how to resume operations after a sympathetic suspension that is caused by an error.

An error occurs on a secondary device for session TUCSON. Session PHOENIX suspends all volumes to the msession recoverable time.

Result: This is an example of the message that was generated:

```
ANTX5104E XRC SESSION(TUCSON) ENCOUNTERED AN ERROR PROCESSING
VOLUME(SEC001) IN VOLUME PAIR(PRIM01,SEC001), RC=613 REAS=0

ANTV8107I SUSPEND COMPLETE FOR VOLUME PAIR(PRIM01,SEC001) FOR SESSION(TUCSON)
  AT CONSISTENCY_GROUP TIME(2004.080 ANTV8107I (CONT) 22:33:21.680331)

ANTV8107I SUSPEND COMPLETE FOR VOLUME PAIR(PRIM02,SEC002) FOR SESSION(TUCSON)
  AT CONSISTENCY_GROUP TIME(2004.080 ANTV8107I (CONT) 22:33:21.680331)

ANTX8120I ALL VOLUMES IN SESSION(TUCSON) ARE NOW SUSPENDED

ANTC8402W XRC SESSION(PHOENIX) COUPLED TO MSESSION(CXRC) DETECTED
  HOLD_STATUS CONDITION IN MSESSION(CXRC),
ANTC8402W (CONT) SUSPENDING ALL VOLUMES

ANTV8107I SUSPEND COMPLETE FOR VOLUME PAIR(PRIM03,SEC003) FOR SESSION(PHOENIX)
 AT CONSISTENCY_GROUP TIME(2004.080 ANTV8107I (CONT) 22:33:24.429661)

ANTV8107I SUSPEND COMPLETE FOR VOLUME PAIR(PRIM04,SEC004) FOR SESSION(PHOENIX)
  AT CONSISTENCY_GROUP TIME(2004.080 ANTV8107I (CONT) 22:33:24.429661)

ANTX8120I ALL VOLUMES IN SESSION(PHOENIX) ARE NOW SUSPENDED
```

Perform the following steps to resume operations after a sympathetic suspension:

1. Generate an XQUERY MASTER report for the master session named CXRC, with a default master high-level qualifier of SYS1 by entering:

```
XQUERY CXRC MASTER
```

The report indicates the status of the XRC sessions coupled to the master session.

Result: The output from this report allows you to determine whether an error has occurred in the coupled session named TUCSON. It also shows the HOLD status of the master session.

```
ANTQ8300I XQUERY STARTED FOR MSESSION(CXRC) MHLQ(SYS1)
ANTQ8202I XQUERY MASTER REPORT - 002
ANTQ8302I SESSION   STA VOL INT CMD   JOURNAL DELTA      RCV/ADV DELTA
ANTQ8303I -------------------------------------------------------------
ANTQ8304I TUCSON    ACT ERR  Y       =00:00:00.000000  +00:00:02.749330
ANTQ8304I PHOENIX   ACT SYM  Y       +00:00:01.253732  =00:00:00.000000
ANTQ8305I TOTAL=2  ACT=2  SUS=0  END=0  ARV=0  RCV=0  UNK=0
ANTQ8306I MSESSION STATUS=HOLD
ANTQ8308I MSESSION RECOVERABLE TIME(2004.080 22:33:24.429661)
ANTQ8309I INTERLOCKED=2  NON-INTERLOCKED=0
ANTQ8301I XQUERY MASTER REPORT COMPLETE FOR MSESSION(CXRC)
```

2. Correct the problem for XRC session TUCSON.

   In this case, the RC=613 on the ANTX5104E message indicates that a hardware error was detected writing to a secondary device. Determine the cause of the problem so that you can resolve it.
3. Update the secondary volumes of the session named TUCSON to msession recoverable time, that is, to a time that is consistent with the volumes of the session named PHOENIX. Issue the following command:

```
XADVANCE TUCSON
```

4. Back up secondary devices for both sessions.

   You can perform your installation backup procedures to back up the data on your secondary devices for both sessions. Both sessions are consistent to the msession recoverable time.
5. Generate an XQUERY MASTER report for the master session named CXRC, with a default master high-level qualifier of SYS1. Issue the following command:

```
XQUERY CXRC MASTER
```

Result: The following output is generated:

```
ANTQ8300I XQUERY STARTED FOR MSESSION(CXRC) MHLQ(SYS1)
ANTQ8202I XQUERY MASTER REPORT - 002
ANTQ8302I SESSION   STA VOL INT CMD   JOURNAL DELTA      RCV/ADV DELTA
ANTQ8303I -------------------------------------------------------------
ANTQ8304I TUCSON    ACT ERR  Y       =00:00:00.000000  +00:00:00.000000
ANTQ8304I PHOENIX   ACT SYM  Y       +00:00:01.253732  =00:00:00.000000
ANTQ8305I TOTAL=2  ACT=2  SUS=0  END=0  ARV=0  RCV=0  UNK=0
ANTQ8306I MSESSION STATUS=HOLD
ANTQ8308I MSESSION RECOVERABLE TIME(2004.080 22:33:24.429661)
ANTQ8309I INTERLOCKED=2  NON-INTERLOCKED=0
ANTQ8301I XQUERY MASTER REPORT COMPLETE FOR MSESSION(CXRC)
```

6. Issue the XCOUPLE command with the RELEASE option to release the master session named CXRC, with a default master high-level qualifier of SYS1, from the HOLD status.

```
XCOUPLE CXRC RELEASE SYS1
```

See "What to do if the master session is in coupled HOLD status" on page 228 for additional information regarding the HOLD status for a master session.

7. Issue the following command to generate a report for the master session CXRC that shows the changed status. After the RELEASE function has completed, the recoverability status of each session is changed to **N** (meaning coupled NON-INTERLOCKED status).

```
XQUERY CXRC MASTER
```

Result: The XQUERY MASTER command generates the following output for the master session CXRC, with the change in status displayed:

```
ANTQ8300I XQUERY STARTED FOR MSESSION(CXRC) MHLQ(SYS1)
ANTQ8202I XQUERY MASTER REPORT - 002
ANTQ8302I SESSION   STA VOL INT CMD   JOURNAL DELTA      RCV/ADV DELTA
ANTQ8303I -------------------------------------------------------------
ANTQ8304I TUCSON    ACT AVS  N
ANTQ8304I PHOENIX   ACT AVS  N
ANTQ8305I TOTAL=2  ACT=2  SUS=0  END=0  ARV=0  RCV=0  UNK=0
ANTQ8308I MSESSION RECOVERABLE TIME(NO_TIME_AVAILABLE)
ANTQ8309I INTERLOCKED=0  NON-INTERLOCKED=2
ANTQ8301I XQUERY MASTER REPORT COMPLETE FOR MSESSION(CXRC)
```

8. Issue the following command to add all suspended volumes back into the coupled session named TUCSON:

```
XADDPAIR TUCSON SUSPENDED
```

9. Issue the following command to add all suspended volumes back into the coupled session named PHOENIX:

```
XADDPAIR PHOENIX SUSPENDED
```

10. Issue the following command:

```
XQUERY CXRC MASTER
```

Result: After you add the volume pairs again, the recoverability status is changed to **Y** (sessions are now interlocked), as displayed by the XQUERY MASTER output for master session CXRC with a default master high-level qualifier of SYS1.

```
ANTQ8300I XQUERY STARTED FOR MSESSION(CXRC) MHLQ(SYS1)
ANTQ8202I XQUERY MASTER REPORT - 002
ANTQ8302I SESSION   STA VOL INT CMD   JOURNAL DELTA      RCV/ADV DELTA
ANTQ8303I -------------------------------------------------------------
ANTQ8304I TUCSON    ACT      Y       =00:00:01.830727  =00:00:00.000000
ANTQ8304I PHOENIX   ACT      Y       +00:00:00.000000  +00:00:02.352970
ANTQ8305I TOTAL=2  ACT=2  SUS=0  END=0  ARV=0  RCV=0  UNK=0
ANTQ8308I MSESSION RECOVERABLE TIME(2004.080 22:43:21.526671)
ANTQ8309I INTERLOCKED=2  NON-INTERLOCKED=0
ANTQ8301I XQUERY MASTER REPORT COMPLETE FOR MSESSION(CXRC) MHLQ(SYS1)
```

Using the XCOUPLE RELEASE command

You can use the XCOUPLE RELEASE command to remove the coupled HOLD status for a master session. In addition, the coupled status of each session is changed to NON-INTERLOCKED. Use caution when you issue this command.

The following is a list of cases where you would use the XCOUPLE RELEASE command:

- After you issue the XQUERY command with the MASTER option.
- After you issue the XADVANCE command for all sessions that do not indicate an RCV/ADV DELTA of zero (0) on the XQUERY MASTER report. The XADVANCE command recovers the data for these sessions to the master recoverable time.
- After you perform manual backups of secondary volumes as required by your system administrator.

In addition, if you issue the XCOUPLE command with either the DELETE or PURGE parameter for the last coupled session in a master session, XRC removes the coupled HOLD status for the master session.

If you couple the XRC sessions through a cluster session, and the cluster session is currently inactive, restart the cluster session. Be sure to do this before you run any XCOUPLE RELEASE command for any currently active XRC sessions that are already coupled through the cluster session. After the cluster session is restarted, issue the XCOUPLE ADD command for each of the sessions that were coupled through the cluster session. After the sessions are recoupled, issue the XCOUPLE RELEASE command. To determine the XRC sessions that have been previously coupled through the cluster session, issue the XSTATUS CLUSTER command.

For additional information about using the XCOUPLE RELEASE command, refer to “Using the XCOUPLE RELEASE command.”

For additional information about restart the cluster session, refer to “What to do if cluster session is in INACTIVE status” on page 234.

XCOUPLE RELEASE command considerations

When you issue the XCOUPLE RELEASE command, XRC removes the coupled HOLD status of the master session. As a result, all coupled sessions in the master session remain coupled to this master session, and the coupled status for each session changes to NON-INTERLOCKED. You can then issue the XADDPAIR command (to add volume pairs to coupled sessions) and the XCOUPLE ADD command (to add new sessions to the master session). In addition, any update writes to primary volumes of coupled sessions changes the master recoverable time.

Attention: After you have issued the XCOUPLE command with either the RELEASE, DELETE, or PURGE parameter, data for the specified session may not be consistent with the other sessions in the master session.

For considerations when XRC sessions are coupled througha cluster session, refer to "XCOUPLE RELEASE command considerations" on page 233.

What to do if a session is in COUPLE_FAILED status

The XCOUPLE_FAILED status for a session occurs when a coupled XRC session is not able to read or write to the master data set because of an error. XRC issues a message to indicate the particular error that occurred. All volumes in the session are suspended. The session remains coupled to the master session. For XRC sessions that are coupled through a cluster session, if one or more of these sessions encounters this type of error, the cluster session indicates to the master session that the cluster session is in a XCOUPLE_FAILED status.

Removing the XCOUPLE_FAILED status for a session

The master data set must be fixed in order to remove this XCOUPLE_FAILED status. Issue the XCOUPLE ADD command for the session after the master data set has been fixed. The XCOUPLE_FAILED status is then removed for the session.

When you issue the XCOUPLE DELETE command, XRC removes the XCOUPLE_FAILED status for the session. This command uncouples the session from the master session. You might need to use the XCOUPLE DELETE command if you cannot fix the master data set for the session.

Removing the XCOUPLE_FAILED status for clustered sessions

If a cluster session terminates for any reason, XRC sessions coupled through a cluster session that indicate a COUPLE_FAILED status show a session status of UNK on the XSTATUS CLUSTER report. To remove the XCOUPLE_FAILED status for clustered sessions, do the following steps:

1. Restart the cluster session.
2. Recouple each XRC session through the XCOUPLE ADD command.
3. After you recouple, use XCOUPLE DELETE command or XCOUPLE PURGE command if you need to run the sessions.

What to do if cluster session is in INACTIVE status

The XSTATUS CLUSTER command provides the current status of a cluster session. If the status of the cluster session is INACTIVE, do the following steps:

1. Restart the cluster session. Use MVS MOFIFY with CREFRESH. If this fails, use the command with the FORCE option. The commands are illustrated below:
   ```
   F ANTAS000,CREFRESH
   F ANTAS000,CREFRESH FORCE
   ```
2. Recouple the XRC sessions through the cluster session.

Removing coupled sessions

To remove coupled sessions from a master session, use the XCOUPLE DELETE command and the XCOUPLE PURGE command.

Using the XCOUPLE DELETE and XCOUPLE PURGE commands

The XCOUPLE DELETE command removes an active coupled session from a master session independent of the status of the coupled session. You must issue the XCOUPLE DELETE command from the logical partition in which the session is currently active. If an XQUERY MASTER command still indicates that the session is coupled to the master session after you have issued the XCOUPLE DELETE command, issue the XCOUPLE PURGE command.

The XCOUPLE PURGE command removes an inactive coupled session from a master session. You can issue the XCOUPLE PURGE command from any logical partition that has access to the associated master data set. XCOUPLE PURGE processing ensures that the session is not active in another logical partition in the same GRS ring.

The XCOUPLE DELETE and XCOUPLE PURGE commands might fail to delete the COUPLE member of the session's state data set because of an error. In this case, you will have to delete the COUPLE member using a data set utility program if you intend to add the session as a coupled session again.

Attention: After you have issued the XCOUPLE command with either the RELEASE, DELETE, or PURGE parameter, data for the specified session may not be consistent with the other sessions in the master session.

Generating dumps of address spaces

Use these procedures to help you generate the following address space dumps:

- Multiple dumps in a single logical partition
- Dumps across multiple logical partitions

Steps for generating multiple dumps in a single LPAR

With the advent of CXRC, you may need to generate dumps of multiple ANTAS*nnn* address spaces at the same time. This cannot be done using a sequence of modify dump commands to the ANTAS*nnn* address space, because the subsequent dumps are not generated until the first one is finished, and they would not be done on a timely basis.

Use the following procedure to generate multiple dumps in a single LPAR:

1. Enter the following command from the console:

   ```
   DUMP COMM=(MULTI EXAMPLE)
   ```

   where *MULTI EXAMPLE* can be any identifying text.
2. In response to the following prompt:

   ```
   xx IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
   ```

   Enter the following command:

```
R xx,JOBNAME=(ANTAS*, 'ANTCL*'),DSPNAME=('ANTAS*'.SYSANT*,'ANTCL*'.SYSANT*),CONT
```

3. In response to the following prompt:

```
yy IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
```

Enter the following command:

```
R yy,SDATA=(ALLNUC,PSA,SQA,CSA,LPA,TRT,SUM,LSQA,RGN),END
```

If you are generating several dumps, and do not want to specify the parameters repeatedly, use the CHNGDUMP command (abbreviated as *CD*). Use *CD* to change the values for the SDUMP dump type (which is used by XRC) to enter the following command:

```
CD SET,SDUMP=(ALLNUC,PSA,SQA,CSA,LPA,TRT,SUM, LSQA, RGN)
R n1,JOBNAME=(ANTAS*, 'ANTCL*'),CONT
R n2,DSPNAME=('ANTAS*'.SYSANT*,'ANTCL*'.SYSANT*),END
```

To put commands in a test case to automatically generate a dump of multiple address spaces at a certain interval, use the following EXPANDER command sequence:

```
CONSOLE CD SET,SDUMP=(ALLNUC,PSA,SQA,LPA,TRT,SUM,LSQA,RGN)
REPLY IEE094D,*,1,* JOBNAME=(ANTAS*),CONT
REPLY IEE094D,*,2,* DSPNAME=('ANTAS*'.SYSANT*)END
WAITFOR ANTC8402W 1 (or whatever message signals the condition, omit if not needed)

CONSOLE DUMP COMM=(ANTC8402W)
```

Generating dumps across multiple LPARs

Note: The above DUMP command options can be specified within the REMOTE parameter of the DUMP command if you want to generate a dump of all XRC address spaces in a sysplex. Use the SYSLIST subparameter to list the names of the different systems to which you want the dump command sent.

The syntax for the REMOTE parameter is:

```
REMOTE(SYSLIST=(sysname1('ANTAS*'),sysname2
 ('ANTAS*')),DSPNAME=('ANTAS*'.SYSANT*))
```

You must specify the PROBDESC=SYSDLOCL parameter when you specify the REMOTE parameter (as indicated in the following example). If you specify the name of the local system in the SYSLIST list, two dumps of the local system will be generated. Therefore, specify the dump options for the local system, and then specify the dump options for the remote systems.

The following example lists all responses that are needed to generate a dump of the local system, as well as the two other systems:

```
DUMP COMM=(DUMP LOCAL AND 2 OTHER SYSTEMS)
R n1,JOBNAME=(ANTAS*), DSPNAME=('ANAS*'.SYSANT*),CONT
R n2,SDATA=(ALLNUC,PSA,SQA,CSA,LPA,TRT,SUM,LSQA,RGN),CONT
R n3,PROBDESC=SYSDLOCL,CONT
R n4,REMOTE=(SYSLIST=(sysname2('ANTAS*'),sysname3
 ('ANTAS*')),DSPNAME,SDATA),END
```

Applying CXRC operational scenarios

The scenarios in this section describe possible ways to manage your CXRC sessions to perform or accommodate various planned data management situations. The following CXRC scenarios are included:

Topic . . .
"Example: recovering from a disaster using the XRECOVER command"
"Example: applying software maintenance in a coupled environment" on page 238

You need to be familiar with the commands that are used in these scenarios. For additional information about the XRC commands, refer to Chapter 5, "Extended remote copy command descriptions," on page 69.

Example: recovering from a disaster using the XRECOVER command

The following scenario describes how to use the XRECOVER command to recover from a disaster. All volumes in sessions coupled to the master session have been suspended as a result of the disaster.

Perform the following steps to suspend coupled sessions and to bring data on the sessions to a consistent, recovered state using the XRECOVER command:

1. Generate an XQUERY MASTER report for the master session named CXRC, with a default master high-level qualifier of SYS1 by entering:

   ```
   XQUERY CXRC MASTER DATASET(RPT1) DISP(OLD)
   ```

   The report indicates the status of the XRC sessions that are coupled to the master session. This allows you to determine that the XRC sessions TUCSON and PHOENIX are still active, but have an error status due to the disaster that has occurred. It also shows the status of the master session, which is HOLD.

   Result: An excerpt of the report is shown below. For an example of the entire report, see "XQUERY MASTER report" on page 191.

```
ANTQ8300I XQUERY STARTED FOR MSESSION(CXRC) MHLQ(SYS1)
ANTQ8202I XQUERY MASTER REPORT - 002
ANTQ8302I SESSION   STA VOL INT CMD   JOURNAL DELTA      RCV/ADV DELTA
ANTQ8303I -------------------------------------------------------------
ANTQ8304I TUCSON    ACT SYM  Y       +00:00:03.824936  =00:00:00.000000
ANTQ8304I PHOENIX   ACT ERR  Y       =00:00:00.000000  +00:00:02.347483
ANTQ8305I TOTAL=2  ACT=2  SUS=0  END=0  ARV=0  RCV=0  UNK=0
ANTQ8306I MSESSION STATUS=HOLD
ANTQ8308I MSESSION RECOVERABLE TIME(2004.080 22:35:21.643364)
ANTQ8309I INTERLOCKED=2  NON-INTERLOCKED=0
ANTQ8301I XQUERY MASTER REPORT COMPLETE FOR MSESSION(CXRC)
```

After the report has been generated, XRC directs the XQUERY output to the data set named RPT1 after first clearing the data set.

2. Issue the following command to suspend the XRC session named TUCSON:

```
XSUSPEND TUCSON TIMEOUT(STORAGE_CONTROL_DEFAULT)
```

3. Issue the following command to suspend the XRC session named PHOENIX:

```
XSUSPEND PHOENIX TIMEOUT(STORAGE_CONTROL_DEFAULT)
```

Result: The following message is generated:

```
ANTQ8300I XQUERY STARTED FOR MSESSION(CXRC) MHLQ(SYS1)
ANTQ8202I XQUERY MASTER REPORT - 002
ANTQ8302I SESSION   STA VOL INT CMD   JOURNAL DELTA      RCV/ADV DELTA
ANTQ8303I -------------------------------------------------------------
ANTQ8304I TUCSON    SUS SYM  Y                         =00:00:00.000000
ANTQ8304I PHOENIX   SUS ERR  Y                         +00:00:02.347483
ANTQ8305I TOTAL=2  ACT=0  SUS=2  END=0  ARV=0  RCV=0  UNK=0
ANTQ8306I MSESSION STATUS=HOLD
ANTQ8308I MSESSION RECOVERABLE TIME(2004.080 22:35:21.643364)
ANTQ8309I INTERLOCKED=2  NON-INTERLOCKED=0
ANTQ8301I XQUERY MASTER REPORT COMPLETE FOR MSESSION(CXRC)
```

Example: applying software maintenance in a coupled environment

The commands that are listed allow you to suspend coupled sessions so that you can perform maintenance on individual sessions, and then add the individual sessions back to the master session after maintenance has been completed.

The following scenario describes how to perform software maintenance operations in a coupled environment:

1. Suspend all sessions coupled to the master session, which is named CXRC and has a default master high-level qualifier of SYS1, on the day and time indicated by the ATTIME parameter by entering:

```
XSUSPEND CXRC ATTIME(2004.145 18:00:00) TIMEOUT(04.00.00)
```

All sessions are suspended with their active duplex volumes at a consistent time across the coupled sessions. You can now apply whatever maintenance processes you want. It is assumed that maintenance will take no more than four hours to complete.

2. Issue the following command to restart the XRC session named TUCSON. After software maintenance operations have completed, take copies of secondary volumes, if needed, to maintain consistency during the resynchronization process.

```
XSTART TUCSON SESSIONTYPE(XRC) ERRORLEVEL(SESSION)
```

3. Issue the following command to restart the XRC session named PHOENIX:

```
XSTART PHOENIX SESSIONTYPE(XRC) ERRORLEVEL(SESSION)
```

4. Take copies of secondary volumes, if needed, after software maintenance operations have completed. Taking copies maintains consistency during the resynchronization process.

Note: Doing things in the order described in the example allows XRC to start tracking changes to volumes that are not on a 2105 controller.

Chapter 9. Extended remote copy data recovery operations

Extended remote copy (XRC) provides the ability to recover data on the recovery system in the event of a disaster to the primary system.

For additional information about how to use XRC for data migration, refer to Chapter 10, "Migrating data with extended remote copy," on page 251.

In this topic

This topic provides information about data recovery with XRC and includes the following sections:

Understanding the XRC volume synchronization process

Understanding how XRC handles record timestamping and volume synchronization will make your recovery procedures easier to establish and accomplish. This section contains the following information about consistency group time and timestamped writes:

Overview of consistency group time

Keeping active volumes consistent up to a given time is an important function of XRC support. Data integrity depends on ensuring that updates occur on the secondary volumes in the same order as they occurred on the primary volumes. This update sequencing is necessary in order to avoid data integrity problems and data loss. Maintaining data integrity becomes especially critical when multiple applications update a volume, or when a data set exists on multiple volumes that are spread across multiple storage controls.

The timestamp reported by the consistency group time is the primary system timestamp up to which the data on XRC active volumes is consistent. The following commands generate messages that report the consistency group time:

- XQUERY
- XDELPAIR
- XSUSPEND
- XRECOVER

- XEND
- XADVANCE

Terms used for consistency group time

The following defined terms are related to consistency group time and are useful to help understand consistency group time. The definitions for these terms refer to the following banking transaction example:

Example: A customer opens an account with a deposit of $400, and then later removes $300. On completion of both transactions, the customer's account shows a balance of $100.

Dependent Write
: A write that should not occur until a previous write operation has completed.

  Example: In the banking example, the withdrawal is dependent on the deposit since the withdrawal would not be possible if the deposit had not occurred. When XRC copies this set of transactions to a secondary set of volumes, XRC must maintain the transaction order since this is a dependent write operation.

Consistent
: XRC guarantees that volumes are consistent when all dependent writes have been secured on the secondary volumes.

  Example: In the banking example, XRC ensures that if the withdrawal transaction is secure on the secondary volume, then XRC has also secured the deposit transaction. There is no condition where XRC would secure the withdrawal without securing the deposit.

Consistency Group
: A collection of application-updated records across all storage controls in an XRC session. XRC forms consistency groups such that records that are dependent on other records (dependent writes) are secured onto the secondary volumes in a consistent way.

  Example: In the banking example, XRC ensures that when a consistency group is generated for the bank transaction, the withdrawal transaction is either in the same group as the deposit transaction or a later group. It will never be in an earlier consistency group. This is true whether XRC secures the data on the same volume, different volumes behind a common storage control, or different volumes behind different storage controls. *This requirement is critical for data base applications which have data spread across numerous volumes on numerous primary storage controls in the configuration*.

Consistency Group Time
: The timestamp to which XRC ensures that XRC volumes are consistent. XRC derives the consistency group time from the system-provided timestamp for all application updates. When applying a consistency group to a set of secondary volumes, XRC reports the latest application timestamp secured on the volumes.

  Example: In the banking example, assume that the deposit and the withdrawal transactions were the only transactions that took place. XRC reports the consistency group time of the withdrawal, as this transaction occurred after the deposit transaction.

Data Exposure
: The approximate time difference (delta time) between data written to the

primary volumes and data secured on the journal data set. The data that is exposed during this time is potentially nonrecoverable. This delta time is zero if all data that has been written to primary volumes for an XRC session has been secured on the journal data set.

Changes to consistency group time

If you periodically issue XQUERY commands to the system, the consistency group time increases as long as the primary volumes continue to have update activity.

If update activity ceases for all volumes in the XRC session, the consistency group time does not change for uncoupled sessions. For coupled sessions, the consistency group time increases. In both cases, an XQUERY report displays an IDLE time that indicates how long application updates have been idle.

The consistency group time reported with XQUERY, XDELPAIR, XSUSPEND, XRECOVER, XEND, and XADVANCE messages is dependent on the status of the XRC volume pair at the time the message is issued. For active XRC volume pairs, the consistency group time reported is the consistency group time for the entire XRC session. For suspended XRC volume pairs, the consistency group time is the time that the volume pair was last consistent with the XRC session.

XRC may suspend a volume pair because of an error, or because you have suspended the volume with an XSUSPEND VOLUME or XSUSPEND TIMEOUT command. XRC reports a consistency group time for the volume pair that reflects the last consistency group that XRC secured on the secondary volume.

Other volume pairs in the XRC session continue to have update activity, and their consistency group time continues to advance. At any given moment, the latest timestamp indicates the last consistency group that XRC has successfully secured on the secondary volumes.

An error can occur while XRC attempts to write records in a consistency group to a secondary volume. XRC may successfully write some of the records in this consistency group before the error occurs. In this case, the consistency group time for that volume pair correlates to the last consistency group that XRC successfully wrote to that secondary volume. If XRC has written all of the records in the consistency group to secondary volumes, then the reported consistency group time reflects the latest timestamp of all the records in the consistency group.

Note:

1. XRC does not assign a consistency group time to a volume in the pending state. The volume has not achieved any consistent state until it is in duplex state. The volume pair is not consistent until XRC has copied all primary volume tracks with their updates to the secondary volume.
2. If none of the volume pairs have been updated since the session was started, XRC reports “NO_TIME_AVAILABLE” for the consistency group time of the volume pair, the session, or both.
3. When the CONTIME parmlib parameters are used with the ANTRQST API to retrieve the consistency time for a session in a coupled master session, the consistency time is the same for all sessions in the coupled master session.
4. Consistency group time is not advanced when L+ volumes are in a session and no I/O activity is seen by XRC for these volumes.

Example: illustration of consistency group time

The example in Figure 10 further illustrates the concept of consistency group time. The frozen point-in-time in the example shows how the system data mover (SDM) processes successive groups of volume updates. The SDM reads data updates from the primary volumes, groups them into time intervals, and then writes them to the journal and secondary volumes. The word `Done` in a column means that the action has completed.

```
Group      Read from       Write to         Write to
            Primary        Journal          Secondary
-----------------------------------------------------------

  1          Done            Done             Done

  2          Done            Done        Not yet started

  3          Done         In process

  4       In process
```

Figure 10. A point-in-time look at the XRC volume synchronization process

The following are definitions for the column headings in Figure 10:

Group This collective information set represents all of the write updates that XRC has received from all of the primary volume's storage controls during a given time interval.

Read from Primary
A "Done" status indicates that the SDM has read the group into its storage.

Write to Journal
A "Done" status indicates that the SDM has successfully written the group onto the journal data sets, which are accessible from the recovery site along with the control data set.

Write to Secondary
A "Done" status indicates that the SDM has completed all update write operations on the secondary disk device for that group's time interval.

Examples: These examples show the consistency group time that is reported for each of the following commands issued at the point in time that is shown in the example in Figure 10.

- If you issue an XQUERY, XSUSPEND, or XEND command, XRC would report the session's consistency group time as the Group 1 timestamp. Group 1 is the last group of updates that XRC successfully wrote to the secondary volumes.
- If you issue an XDELPAIR command to a volume pair that is in the session, XRC reports the consistency group time for the last group of updates that XRC wrote to the secondary volume. If the volume pair is still active in the session, the returned consistency group time is the same as the XRC session consistency group time.
- If you were to issue an XRECOVER command at this point in time, the secondary disk device would be consistent up to the Group 2 timestamp. That is the last completed group of interval data that is secured on the journal. Write updates from Groups 3 and 4 would be lost, and considered "bytes in transit". In this case, the data exposure would be the time delta between Group 4 (data written to the primary) and Group 2 (data secured on the journal). Updates will

only be applied to secondary volumes if the consistency group timestamp is less than or equal to the earliest interlocked session's journal time.

Note: For a coupled session that was not ended or suspended by an XRC command, the starting *consistency_group_time* reported is the timestamp for the last known update for the session when the following conditions exist:
- A session status is noninterlocked
- Updates were not occurring when the session ended (the session was idle).

If the above conditions exist, the consistency time reported may be earlier than the master recovery time or the last session consistency time indicated in the output of an XQUERY command.

Understanding the importance of timestamped writes

You can maximize XRC data consistency when you ensure that all writes to primary volumes are timestamped. MVS device support code automatically timestamps all writes to XRC volumes.

Nontimestamped writes can exist for primary systems that are not at the required software system level. Nontimestamped writes can also exist for applications on non-MVS systems that bypass the device support code by performing direct I/O. In addition, writes to MVS paging datasets are also done without timestamps.

When XRC encounters nontimestamped write information, the following occurs:
- XRC issues an ANTX8030W message after encountering 5000 nontimestamped tracks on active volume pairs.
- The data returned with the XQUERY command shows a seqcheck condition in the status field, which warns you that the last recorded update is a nontimestamped write. The volume is no longer in seqcheck status when the last write of a subsequent consistency group is a timestamped write.

Care should be taken when adding volumes to an XRC session where nontimestamped writes are done to the volumes. If nontimestamped writes are issued to volumes in a session where timestamped writes are scarce, XRC may experience delays in shadowing these updates to secondary volumes. This delay will cause other coupled sessions to experience the same delays. Such delays can result in excessive accumulation of updates in cache, increased application pacing, and increased risk of Long Busy conditions.

If you plan to shadow volumes containing data sets for which writes are issued without a timestamp, consider shadowing these volumes in a separate, uncoupled data mover. Otherwise, ensure that you have other volumes in the session that will provide a continuous source of timestamps.

XRC copies nontimestamped writes to the secondary volume in the same way that it copies timestamped writes. A specific nontimestamped write that is dependent on other writes on other primary volumes may be applied to the secondary volume in an out-of-order sequence.

Example: If a bank account record is updated with a deposit of $400.00 and then updated with a withdrawal of $300.00, the account record would show an increase of $100.00. Assume that the processing system does *not* timestamp these data updates. The system applies the deposit update to the account record milliseconds *after* it applies the withdrawal update from a different volume. Under normal conditions, the volumes at both the primary site and the recovery site receive both updates. It is possible, though, that a disaster at the primary site could destroy the volume that contains the deposit information before the system could add this deposit data to the secondary volume. As a result, the deposit update would not be present on the recovery subsystem. The secondary volume record at the time of recovery would show a *decrease* of $300.00, with no indication that there is missing data.

Timestamped writes allow XRC to guarantee data consistency. To take advantage of the timestamping feature, ensure that all participating host systems operate with the latest device support code.

If the nontimestamped write has no dependencies on other writes, then it does not jeopardize data integrity. In either case, this condition is temporary and corrects itself during normal operations as XRC applies each set of updates to the secondary volume. However, as indicated previously, nontimestamped writes may cause delays in shadowing all updates for an XRC session. In particular, coupled XRC sessions may cause other XRC sessions to experience the same delays.

Recovering data with XRC—Uncoupled and coupled sessions

This section contains the following information about recovering data with XRC:

- "Recovering data with XRC"
- "Creating a recovery volume report" on page 248
- "Advancing consistency time using the XADVANCE command" on page 249

Recovering data with XRC

Before you begin: If the XRC system data mover is running at the recovery site at the time of the failure, first issue an XEND command, then go to step 3.

Perform the following steps to have the recovery system take over for the primary system:

1. Reconfigure the path connections, if necessary, to connect local systems to the recovery system disk.
2. Start the DFSMS address space, if it is not already active on the recovery system. Run the IDCAMS recatalog function before activating the recovery system. The state, control, journal, and master (if applicable) data sets need to be cataloged on the recovery system. If you use a cluster session to couple sessions, catalog the cluster data set on the recovery system.

   If you are running multiple XRC sessions, you must issue the XRECOVER command for each individual session. If the sessions are coupled, the XRECOVER command coordinates the recovery of the sessions so that each session is recovered to the same consistency time. A master XRECOVER command does not exist. If you couple sessions to different cluster sessions, issue the XRECOVER commands for those sessions coupled to one cluster session before you issue the commands for another cluster session.
3. Issue the XRC command XRECOVER for each XRC session on the recovery system to add all valid, nonapplied journal data to the secondary (target) volumes. This action also sets the secondary volume serial numbers equal to the serial numbers of the primary volumes for those volumes which had previously reached a duplex state.

   Rule: The XRC recovery function must have access to the appropriate journal, control, and state data sets that were in use on the recovery system at the time of the failure. When the session is coupled, the recovery function must have access to the master data set. If you couple the sessions by using a cluster session, you must have access to the cluster data set.

The XRECOVER command does not process volumes that are in the pending state, or that were suspended before reaching duplex state, as they have never reached duplex state. XRC recovery will not apply data to secondary volumes if the session was ended or suspended by command. The exception to this is when an error occurs during an update to a secondary volume while there is data from the journal waiting to be written to the secondary volume. This allows the volume to be recovered to the same consistency time as the rest of the secondary volumes in the session.

Note: For a coupled session that was not ended or suspended by an XRC command, the starting *consistency_group_time* reported is the timestamp for the last known update for the session when the following conditions exist:

- A session status is noninterlocked
- Updates were not occurring when the session ended (the session was idle).

If the above conditions exist, the consistency time reported may be earlier than the master recovery time or the last session consistency time indicated in the output of an XQUERY command.

If XRC locates valid updates to suspended volumes, XRC applies those updates to the secondary volumes and increments their suspension times accordingly. If the volume suspension time is different than the recovery time of the XRC session, the data on the volume may not be consistent with data on other volumes.

The XRC recovery function uses the appropriate journal, control, and state data sets to put all secondary volumes in a known consistent state. The XRECOVER command, as part of XRC's recovery function, automatically creates an XQUERY volume report to assist you with recovery (see "Creating a recovery volume report" on page 248 for more information). XRC also initiates a request sequence to vary the secondary volumes offline, and then online to bring them to a ready state.

Note: In an actual disaster recovery, the primary volume would be offline when you issue the XRECOVER command. You must manually vary the primary volume offline during a disaster recovery verification test.

4. Change the volume serial numbers, if needed. You might notice that some recovery system disk devices may still be in an error state after you have completed the preceding steps. Changing the volume serial numbers prevents the application from accessing the volume until recovery procedures have been restored or have updated their contents.

   Note: It is not enough to vary these devices offline, as other applications can bring the devices online to another system.
   If the secondary has more cylinders than the primary, run the ICKDSF REFORMAT REFVTOC function to the secondary volume from a system in the production SMSPlex to refresh the VTOC and reflect the additional space, and to ensure that DFSMS space statistics accurately reflect the space on the secondary volume.

   You might occasionally find that you cannot access data on a successfully recovered volume because the volume's indexed VTOC has been disabled. In this case, run the BUILDIX function of ICKDSF to enable the indexed VTOC.

5. Restart primary systems and perform the same systems and application startup procedures that are performed on the primary system when applications start up following a system failure.

You might want to include catalog volumes as part of the data that you copy to the recovery system. Recovery system catalog entries will then be consistent with those on the primary system. Use the procedures that are listed in “Copying the catalog and control data sets” on page 62 to manage catalog updates that are not made to the recovery system.

Creating a recovery volume report

The XRECOVER command, as part of the XRC recovery function, automatically generates an XQUERY volume report to assist you with recovery.

Example: The following is an example of an XQUERY volume report that was issued by an XRECOVER command. See “XRECOVER–Recovering data on the recovery system” on page 100 for further information on the XRECOVER command.

When the XRECOVER SESSION1 HLQ(SYS1) command processes, this report is generated:

```
ANTQ8200I XQUERY STARTED FOR SESSION(SESSION1) ASNAME(ANTAS001)
ANTQ8202I XQUERY RECOVER REPORT - 002
ANTQ8271I ------ORIGINAL-------
ANTQ8274I PRIMARY  SECONDARY STA CON CLP  ------TIMESTAMP---------
ANTQ8203I -------------------------------------------------------
ANTQ8275I PRIM01   SECD01    DUP YES YES
ANTQ8275I PRIM02   SECD02    DUP YES YES
ANTQ8275I PRIM03   SECD03    SUS NO  YES  2000.322 19:54:00.562356
ANTQ8275I PRIM04   SECD04    PND NO  NO
ANTQ8237I TOTAL=4   DUP=2   PND=1   SUS=1
ANTQ8231I DATA CONSISTENT(2000.322 20:55:00.356210)
ANTQ8232I SESSIONTYPE(XRC)  ERRORLEVEL(SESSION)  HLQ(SYS1)
ANTQ8201I XQUERY RECOVER REPORT COMPLETE FOR SESSION(SESSION1)
```

The XADVANCE command is also part of the XRC recovery function. The XADVANCE command differs from the XRECOVER command in that it applies updates as the XRECOVER command does, however, it does not *clip* the volumes.

Example: The following is an example of an XQUERY ADVANCE report. Issuing an XADVANCE command to a master session automatically generates an XQUERY MASTER report for the coupled sessions. See “XADVANCE—Updating secondary volumes” on page 77 for further information on the XADVANCE command.

When the XADVANCE SESSION1 HLQ(SYS1) command processes, this report is generated:

```
ANTQ8200I XQUERY STARTED FOR SESSION(SESSION1) ASNAME(ANTAS001)
ANTQ8202I XQUERY ADVANCE REPORT - 001
ANTQ8272I PRIMARY  SECONDARY STA CON  ------TIMESTAMP---------
ANTQ8203I --------------------------------------------------------
ANTQ8273I PAS001   SES001    SUS YES  2000.322 20:55:00.356210
ANTQ8273I PAS002   SES002    SUS YES  2000.322 20:55:00.356210
ANTQ8273I PAS003   SES003    SUS NO   2000.322 19:53:00.324610
ANTQ8237I TOTAL=3   DUP=0   PND=0   SUS=3
ANTQ8231I DATA CONSISTENT(2000.322 20:55:00.356210)
ANTQ8232I SESSIONTYPE(XRC)  ERRORLEVEL(SESSION)  HLQ(SYS1)
ANTQ8201I XQUERY ADVANCE REPORT COMPLETE FOR SESSION(SESSION1)
```

The XRECOVER and XADVANCE reports are similar to a regular XQUERY volume report. Volume pairs that were in duplex state at the time of recovery have had all journal data applied to their secondary volumes. The report shows active volumes that are in pending state and suspended volumes that are in pending state.

For additional information about the XRC messages that are listed within the XRECOVER and XADVANCE reports, refer to *z/OS MVS System Messages, Vol 1 (ABA-AOM)*.

Advancing consistency time using the XADVANCE command

The XADVANCE command can be used to recover secondary volumes to a consistent time. The difference between the XRECOVER command and the XADVANCE command is that the XADVANCE command does not change the volume serial labels on the secondary volumes. The XADVANCE command can be used if you wish to bring all volumes in the XRC session to a consistent state in order to make a copy of secondary volumes before restarting a session. However, for the XRECOVER command, XRC recovery will not apply data to secondary volumes if the session was ended or if the volume was suspended by command.

The exception to this is when an error occurs during an update to a secondary volume while there is data from the journal waiting to be written to the secondary volume. This allows the volume to be recovered to the same consistency time as the rest of the secondary volumes in the session.

Note: For a coupled session that was not ended or suspended by an XRC command, the starting *consistency_group_time* reported is the timestamp for the last known update for the session when the following conditions exist:

- A session status is noninterlocked
- Updates were not occurring when the session ended (the session was idle).

If the above conditions exist, the consistency time reported may be earlier than the master recovery time or the last session consistency time indicated in the output of an XQUERY command.

Table 35 on page 250 outlines how updates occur when you issue the XADVANCE command for active and inactive sessions.

Table 35. Using the XADVANCE command

If the session is . . .	Then . . .
Active	All volumes must be suspended by error or by command. If volumes are suspended because of an error, issue the XADVANCE command to bring those volumes to a consistent state. The command performs the following updates depending on whether the session is coupled or uncoupled: • If the session is not coupled, this command applies all updates that have been written to the journal and have not been written to the secondary volumes. • If the session is coupled, this command applies all updates that have been written to the journal and have not been written to the secondary volumes that have timestamps less than or equal to the master session's master recoverable time.
Inactive	The session has been suspended or has ended. If volumes are suspended because of an error, issue the XADVANCE command to bring those volumes to a consistent state. The command performs the following updates depending on whether the session is coupled or uncoupled: • If the session is not coupled, this command applies all updates that have been written to the journal and have not been written to the secondary volumes. • If the session is coupled, this command applies all updates that have been written to the journal and have not been written to the secondary volumes that have timestamps less than or equal to the master session's master recoverable time.

Chapter 10. Migrating data with extended remote copy

The purpose of an XRC migration session is to migrate volumes from one location to another. The migration can be between local devices or between remote sites. The XRC migration session ensures that data for all volumes is consistent when the migration is complete. It is not important to be able to recover the data at all times. A typical migration is of relatively short duration.

Note: The XCOUPLE ADD command fails if you specify SESSIONTYPE(MIGRATE) on the XSTART command. If you restart a coupled session and you specify SESSIONTYPE(MIGRATE), the MIGRATE option will be ignored.

XRC might be the best choice for moving your data to other disk devices, depending on the device types of the primary and secondary volumes.

In this topic

The following section are included in this topic:

Section . . .
"Steps for migrating data with XRC"
"Migration operation considerations" on page 252

Steps for migrating data with XRC

Before you begin, you need to be familiar with the commands used in the following steps. For additional information about XRC commands, refer to Chapter 5, "Extended remote copy command descriptions," on page 69.

The steps for data migration with XRC are as follows:

1. Initiate a migration session by specifying SESSIONTYPE(MIGRATE) on the XSTART command. The XSTART command directs the system data mover to start a migration session. (XRC uses the system data mover for disaster recovery purposes and for migration.)

   In migration mode, data consistency across all volumes is only important when migration is complete. Therefore, it is best to specify ERRORLEVEL(VOLUME) on the XSTART command for a migration session. If an error occurs, XRC suspends the volume pair that incurred the error. After you correct the error, issue an XADDPAIR SUSPENDED command to add back the volume pair into the XRC session. Meanwhile, XRC continues to copy other volume pairs.

   To start a session called, for example, MIGRATE, you must already have allocated the state data set with the session ID of MIGRATE. See "Specifying XRC journal, control, and state data sets" on page 54 for more information.

   Example: The following example starts a session named MIGRATE:

```
XSTART MIGRATE SESSIONTYPE(MIGRATE) ERRORLEVEL(VOLUME)
```


2. Establish an XRC pair with an XADDPAIR command. Use the default ERRORLEVEL(VOLUME) option, and do *not* specify NOCOPY. XRC issues an ANTI8024I message when the volume or volumes are in the duplex state. This message means that the initial copy is complete and that XRC has updated the secondary volumes with all of the changes that have occurred during the copy operation.
3. Monitor the initial copy process by issuing XQUERY TSO commands. The XQUERY command returns the percentage of the initial copy that has completed. During the initial copy, XRC also copies all primary volume updates to the secondary (target) volume.
4. Verify that the XRC volumes are in duplex state with an XQUERY command. When the volume pair attains duplex state, XRC has copied all of the primary volume contents and all subsequent updates to the XRC secondary volume. XRC continues to update the secondary volume as changes occur to the primary.
5. Switch to the secondary (target) volume when the copy operation is complete. To switch your application from the primary volume to the secondary volume, do the following:
   a. Stop all applications that are using the primary volume.
   b. To complete the migration at a known point in time, issue the XEND command with the ATTIME keyword or the DRAIN keyword. Specify ATTIME to ensure that XRC does not copy any updates to secondary volumes beyond the specified timestamp. Specify DRAIN when you have quiesced primary application activity and want to ensure that all application I/O has been applied to the secondary volumes before ending the session. Issue the XEND(DRAIN) command to secure all updates onto the XRC secondary volume. When all application updates have stopped, no further updates occur on the primary volumes.
   c. Issue an XRECOVER command to change the secondary volume serial number to be the same as the primary volume serial number.
   d. Vary the original primary device offline. If the secondary volume contains more cylinders than the primary volume, you must run ICKDSF Release 17 (or above) with the REFORMAT REFVTOC command against the secondary volume. This action refreshes the VTOC to reflect the additional space. The ICKDSF function should be run from a system in the production SMSPlex to ensure that DFSMS space statistics accurately reflect the larger secondary volume. If you do not run ICKDSF with REFORMAT REFVTOC, the system cannot recognize or use the extra space on the secondary volume.
   e. Restart all applications, which will now use the new disk volumes.

Migration operation considerations

The following considerations can affect migration mode operations:

- A migration session does not require that you allocate control data sets or journal data sets. However, it does require the state data set. Because there are no control or journal data sets, all data is consistent on the secondary volumes up to the timestamp reported when the session is ended.

  Note: Migration mode uses fewer system resources than disaster recovery mode because XRC does not write data to the journal data sets or control data sets. The primary application update rate supported in migration mode is about the same as in disaster recover mode.

- To complete the migration faster, you may want to increase the number of concurrent synchronization tasks. To do this, use the XSET command to:
  - Change the SYNCH value to increase the total number of concurrent tasks that perform the migration copy.
  - Change the SCSYNCH value to increase the maximum number of tasks that can be active concurrently on a single storage control.

  You might need to allocate additional fixed storage before starting additional tasks. See “XSET–Changing session parameters” on page 102 for the XSET command syntax.
- In migration mode, the SDM copies track images from the primary volume to the secondary volume. Concurrently, record updates made to track images, which have already been copied, are applied to secondary volumes. The record updates are read in the same way that SESSIONTYPE=XRC updates are processed, namely, by utility devices. The requirements for SESSIONTYPE=XRC apply to SESSIONTYPE=MIGRATE. There must be a utility device that is defined for each volume pair.

  Recommendation: In a channel-extender environment, define utility devices with the XSET UTILITY(FIX) command. If the migration is done during times of heavy stress on the system, sufficient buffer space must be available for the SDM. Specify the PAGEFIX parameter to provide a buffer so that minimal application system impact is experienced.
- You can change the session type whenever you issue the XSTART command. Initially, you may start a migration session to copy all volumes from one location to another. When the copy has completed, you may then choose to suspend the XRC session and restart the session as a disaster recovery session. Remember that you must define journal data sets and control data sets for a disaster recovery session.

  Note: The SESSIONTYPE of MIGRATE cannot be used for coupled XRC sessions.

For an example of a migration scenario, see “Example: migrating data” on page 196.

Chapter 11. Recovering from error conditions using extended remote copy

This chapter describes various error conditions and how to recover from them.

In this topic

The following subjects are included in this topic:

Specifying the ERRORLEVEL parameter in coupled and uncoupled sessions

The actual XRC error recovery process depends on both of the following conditions:

- The specific failure condition **and**
- The error level that is specified for that volume pair.

You can specify the default error level for the entire session when you start the XRC session by issuing the XSTART command and specifying the ERRORLEVEL parameter. You can also change the error level for specific volume pairs when you issue the XADDPAIR command with the ERRORLEVEL parameter specified. The ERRORLEVEL parameter aids in the error recovery process. It does not cause session errors.

To specify how XRC processes certain error conditions, use one of the following ERRORLEVEL subparameters:

- VOLUME
- SESSION
- *group_name*

For additional information about the XSTART and XADDPAIR commands and their parameters, refer to "XSTART–Starting a session" on page 110 and "XADDPAIR–Adding volume pairs or utility volumes" on page 71.

Comparing the ERRORLEVEL options

Table 36 on page 256 compares the different options of the XADDPAIR ERRORLEVEL parameter and describes how different system failures affect the suspension of XRC volume pairs. "Active volumes" refers to volumes that are not in suspended state.

Table 36. XRC system failure volume pair suspensions, determined by ERRORLEVEL options

ERRORLEVEL option	Single volume failure	Primary storage control failure	System data mover failure
(Volume)	One active volume pair is suspended.	All active volumes associated with the storage control session are suspended.	All active volume pairs in the session are suspended.
(Session)	All active volume pairs in the storage control session are suspended. If the session is coupled, all volumes in all other coupled sessions associated with the same master session are also suspended.	All active volumes in the storage control session are suspended. If the session is coupled, all volumes in all other coupled sessions associated with the same master session are also suspended.	All active volume pairs in the XRC session are suspended. If the session is coupled, all volumes in all other coupled sessions associated with the same master session are also suspended.
(*group_name*)	All active volume pairs in the group are suspended.	All active volumes on the storage control are suspended, and all active volumes in this group on any storage control are suspended. See Note.	All active volume pairs in the XRC session are suspended.

Note: A primary storage control error suspends all XRC volumes that are attached to it. If any volumes on that storage control belong to an XRC named group, those volume groups become suspended on other storage controls.

ERRORLEVEL(VOLUME)

The VOLUME option suspends an XRC volume pair that encounters a device error. XRC continues to process all other volume pairs in the XRC session. You must correct the volume's error condition and then resynchronize it so that it is consistent with the other volumes in the XRC session.

If you select VOLUME and the error is associated with the XRC session, then XRC suspends all volume pairs for that storage control. Volume processing continues for other storage controls.

Note: If a volume that is suspended due to an error contains data that is dependent on data on another volume, data consistency cannot be assured in the event of a disaster recovery.

ERRORLEVEL(SESSION)

The SESSION option suspends all volume pairs in the XRC session when an error is associated with any duplex volume that has ERRORLEVEL(SESSION) specified. The system data mover (SDM) then continues to read the updates from the primary storage controls. It does not copy data to secondary volumes until you have added the volume pairs back to the session. You need to copy only the changed data when you add the volumes. You must correct the error condition and resynchronize the suspended volume pairs. If SESSION is selected and a failure occurs on a single storage control that attaches to any duplex volume pair with ERRORLEVEL(SESSION) specified, then all XRC volume pairs are put into suspended state. XRC suspends volume pairs that are in pending state, and the pairs remain where they are in synchronization. Software bitmapping continues for all volumes that are part of the XRC session. In addition, if the affected session is a coupled session, all volumes in all other coupled sessions that are associated with the same master session will also be suspended. The master session is placed into HOLD status to protect the recoverable time for each XRC session.

ERRORLEVEL(group_name)

The *group_name* option allows you to associate any valid TSO name with one or more volume pairs. An error that occurs on a single storage control that attaches to any duplex volume pair that has ERRORLEVEL(*group_name*) specified causes XRC to suspend all other volume pairs that have been assigned the same group name specification. If the same *group_name* is used on other sessions that are coupled in the same master session, the volume pairs in the other sessions are not suspended.

Note: Volumes must be in duplex state before an error that is related to the associated volumes can cause other duplexed volumes to become suspended. This prevents the following error conditions for volume pairs that are still in pending state from suspending all volume pairs in the session:
- Volume serial numbers entered incorrectly
- Track copy errors during volume synchronization
- Spurious track errors during volume synchronization

You can diagnose and correct these XADDPAIR command-related errors independently of other XRC session activity. When you have corrected the error, you can make the volume pair operational by reissuing the XADDPAIR command.

Steps for recovering from environmental errors

Take action that is based on the scope and the location of the error. Automation functions can take action based on the presence of ANTX5xxx Etype errors, which report the error, the cause of the error, and the volumes that are affected.

Before you begin: You need to be familiar with the commands used in these steps.

Use the following procedure to recover from environmental errors:
1. Correct the error.
2. Restart the primary system, if necessary.
3. Issue an XADDPAIR SUSPENDED command to add the suspended volume pairs back to the XRC session. XRC automatically copies all pending updates when the application programs become active.

For additional information about the XRC commands, refer to Chapter 5, “Extended remote copy command descriptions,” on page 69.

Steps for recovering from processor, address space, and XRC data set failures

XRC error conditions may occur as a result of various failures on an application processor. These errors can cancel critical address spaces that reside within the affected system, and may in turn cause the application to end. The system data mover may also be on the system with the failure.

Other error situations may affect only the SDM system. The SDM may be affected by an environmental problem, an internal SDM error, errors on the XRC state, control, or journal data sets, or due to the operator canceling the ANTAS*nnn* address space.

Before you begin: You need to be familiar with the commands used in these steps. For more information, refer to Chapter 5, "Extended remote copy command descriptions," on page 69.

Perform the following steps to recover from processor, address space, and XRC data set failures:

1. Correct the error.
2. Restart the primary system, if necessary.
3. Restart the SDM system, if necessary. For many errors, the XRC session will restart itself.
4. Issue an XSTART command to start the XRC session, if necessary.
5. Issue an XADDPAIR SUSPENDED command to add the suspended volume pairs back to the XRC session. XRC automatically copies all pending updates when the application programs become active.
6. Issue the XCOUPLE *msessionid* RELEASE command, if the XRC session is coupled. To clear up a cluster session, issue the MVS MODIFY command of **F ANTAS000,CREFRESH FORCE**. See Appendix A, "Advanced Copy Services diagnostic aids," on page 549 for a description of this command. You can use the **F ANTAS000,CREFRESH FORCE** command in the following circumstances:
   - The ANTCL*nnn* address space is not active, and you cannot enable the ANTCL*nnn* address space by issuing the CREFRESH command without FORCE.
   - The ANTCL*nnn* address space is active, and you cannot deactivate the ANTCL*nnn* address space by issuing the CREFRESH command without FORCE.
   - The master session consistency time does not advance, and messages identify the cluster as the reason that the master consistency time is delayed.

   In a disaster environment, you might impact data consistency if you incorrectly use the MVS MODIFY command of **F ANTAS000,CREFRESH FORCE**.

Using mirror status verification to check for non-duplex volumes

Mirror status verification function checks for the presence of non-duplex volumes in the session. The function also checks duplex volumes for unexpected XRC device status in the primary storage subsystem. Exceptions are reported in system messages. Duplex volumes found to have unexpected device status are immediately suspended. You can specify the AUTO_READD flag to automatically readd the volumes.

The mirror status verification function is invoked under the following four circumstances:

- When an XQUERY command with the ACTIVITY parameter is used.
- When the VerifyInterval parameter specified in the XRC PARMLIB expires.
- When the data mover detects that a cluster failback has occurred on the primary storage subsystem.
- When the data mover detects that mirroring might have been interrupted for a duplex volume.

The mirror status verification function issues message ANTX8150I - ANTX8153I and might suspend volumes. See *z/OS MVS System Messages, Vol 1 (ABA-AOM)* for more information.

Mirror status verification exploits the recordset accumulator query capability of the primary storage subsystem, if available. To ensure that the data mover recognizes the capability and provides full verification function, issue the following command to each XRC session:

```
XQUERY session_id STO FEATURES
```

If a - is displayed, contact your hardware support representative for information on upgrading microcode to a level that supports write pacing.

Using state saves to diagnose suspected problems

To diagnose suspected problems in XRC, you can direct a storage control to dump its internal control queues. This is called a state save. You can request a standard state save, which is followed by a warmstart, or a *non-disruptive* state save, which does not result in a warmstart. You can also supply additional descriptive information that is stored in the dump.

One statesave is allowed every 5 minutes. However, only 10 non-disruptive statesaves are allowed in a 24-hour period.

You request a state save with keywords in PARMLIB, an operator command or with ANTRQST or ANTTREXX keywords. For more information, refer to:

- “ANTXIN00 parmlib parameters” on page 126
- “STATESAVE operation (XRC)” on page 560
- Appendix C, “ANTRQST and ANTRQSTL macros – call to the system data mover API,” on page 573
- Appendix D, “REXX support for the ANTRQST API,” on page 745.

Part 3. Peer-to-Peer Remote Copy

This provides the information to help you use Peer-to-peer remote copy (PPRC).

Chapter 12. Planning for Peer-to-Peer Remote Copy

Before installing and using PPRC you need to plan for it. Understanding the software and hardware requirements will help you prepare to use PPRC in your installation.

In this topic

The following subjects are included in this topic:

PPRC requirements

You must meet the software and hardware requirements listed in the following sections in order to plan for and successfully install PPRC. Table 3 on page 15 includes additional information about PPRC hardware and software requirements.

PPRC software requirements

z/OS, VM, VSE, and ICKDSF stand-alone systems support PPRC.

z/OS software requirements

PPRC TSO command functions, ANTRQST API and ANTTREXX API requests are supported on all current releases of z/OS.

VM software requirements

VM/ESA 2.1.0 (and above) supports PPRC operations. See the items identified as "For VM Systems" under "PPRC operational considerations" on page 264.

ICKDSF support

ICKDSF release 17 and above supports PPRC operations for z/OS, VM, VSE, and stand-alone platforms. The ICKDSF PPRCOPY command provides functions that correspond to the PPRC MVS TSO command set. Additional PPRC functions not supported by the TSO command set may also be available through ICKDSF.

For additional information about ICKDSF, refer to *Device Support Facilities (ICKDSF) User's Guide and Reference*.

PPRC hardware requirements

The following are PPRC hardware requirements:

- Both primary and recovery site storage controls must be of the same type (peer to peer).

Example: If the primary site storage control is an IBM Enterprise Storage Server (ESS), the recovery site storage control must also be an ESS.

- For ESS, the appropriate features must be enabled and licensed microcode installed.
- PPRC allows the use of Fibre Channel Protocol (FCP) or ESCON connections between your primary and secondary subsystems depending on the capabilities of the storage subsystem. The ESCON connections can be direct fiber optic connections or through ESCON Directors. The FCP connections can also be direct or through fibre channel switch.
- PPRC-capable Licensed Internal Code (LIC) must be installed on both the primary and secondary storage subsystems. The levels of PPRC LIC on the systems must be compatible.
- A compatible secondary volume must be available for each primary volume you need to copy. The secondary volume must have the identical track capacity and number of tracks per cylinder and either the same or larger volume capacity.
- PPRC secondary devices located in an alternate subchannel set must:
  - Have the same 4-digit device number as the associated PPRC primary device in subchannel set 0
  - Be defined as 3390D devices in the I/O configuration using HCD.

PPRC supported devices

PPRC supports all primary and secondary subsystems that have PPRC-capable LIC.

Note: Throughout the following PPRC chapters references to the scope of a command or operation is often identified as applying to a logical subsystem. In the case of storage subsystems which do not have logical subsystems, the scope is the entire physical storage subsystem.

PPRC operational considerations

This section addresses the physical and operational considerations for PPRC. Topics include conditions for establishing PPRC paths and volume format restrictions.

The following considerations apply to PPRC operations:

- Fixed Block (Open) devices in PPRC sessions can be controlled by TSO and ANTRQST commands as long as there is a CKD device configured in the same cluster as the FB devices.
- To ensure data integrity, all CKD volumes that are part of PPRC copy operations must conform to the following volume format, track, and access method restrictions:
  - Volumes must have a standard format for record zero (R0). Volumes with R0 data lengths longer than 8 bytes can cause a track format error to remain undetected when the storage control formats the track in cache.
  - You cannot assign alternate tracks in the user area. User data can be overlaid if you assign a user track on the primary address as an alternate track for a secondary address.
  - All storage control extent specification commands must specify normal access authorization mode. Data written to a PPRC primary device while the storage control is in diagnostic or device support mode is not copied to the PPRC secondary volume. It is therefore important to remove volumes from PPRC pairs before running a utility program like ICKDSF.

- PPRC secondary volumes can only accept a subset of I/O operations while they are secondary devices (much like dual copy volumes). IBM recommends that you vary the attached secondary volumes offline to all attached systems. The PPRC secondary volume serial numbers are the same as those on the primary volumes.
- PPRC allows the mixed use of ESCON and FCP connections.
  - ESCON connections are single directional. You cannot use an adapter as both the source of one path and the destination of another path. But you can establish multiple paths using the adapter as a source, or multiple paths using the adapter as a destination.
  - The FCP physical connection can be used in both directions, and can have many logical paths established on that connection, some going in opposite directions.

  The following table summarizes some of the differences between PPRC FCP and PPRC ESCON technology:

FCP compared to ESCON	FCP	ESCON
PPRC primary port accepts host I/O	Yes	No
PPRC secondary port accepts host I/O	Yes	Yes
Link is full-duplex (have paths established in both directions simultaneously)	Yes	No
Supports synchronous PPRC	Yes	Yes
Supports PPRC Extended Distance	Yes	Yes
Supports Global Mirror	Yes	No
Number of PPRC logical paths defined between the primary and secondary LSSs	8	8
Number of logical paths from a primary ESS (32 PPRC paths between one primary and four secondary ESSs)	1024	1024

- **For VM Systems:** Native VM supports PPRC with VM/ESA 2.1.0 and above. Observe the following when you operate PPRC as a guest under VM/ESA 2.1.0 and above:
  - PPRC volumes must either be dedicated volumes, or you must define them as fullpack minidisks. This includes DEVNO-defined minidisks.
  - The VM guest directory must include an entry that states “STDEVOPT DATAMOVER=YES”.

  For VM/ESA levels prior to 2.1.0, you must define PPRC volumes to VM as unsupported disk, and you cannot define them as minidisks.

Examining PPRC configuration options

Peer-to-peer remote copy allows mixed use of ESCON and FCP types based on the following criteria: ther

- An LSS to LSS pairing — you can only have one type of path, ESCON or FCP.
- Paths for a source or target LSS can be mixed, as long as the different paths do not involve the same source and target LSSs.

You can define from one to eight ESCON or FCP paths, depending on the capability of the logical subsystem, from a single primary site logical subsystem to a specific recovery site logical subsystem. You can also attach up to 16 recovery site logical subsystems to each primary site logical subsystem. Therefore, a total of 128 paths can be defined between a single primary site logical subsystem and 16 recovery site logical subsystems. Figure 11 shows examples of possible ESCON path configurations between PPRC primary and recovery logical subsystems systems. Figure 12 on page 267 shows an example of a configuration using FCP and ESCON.

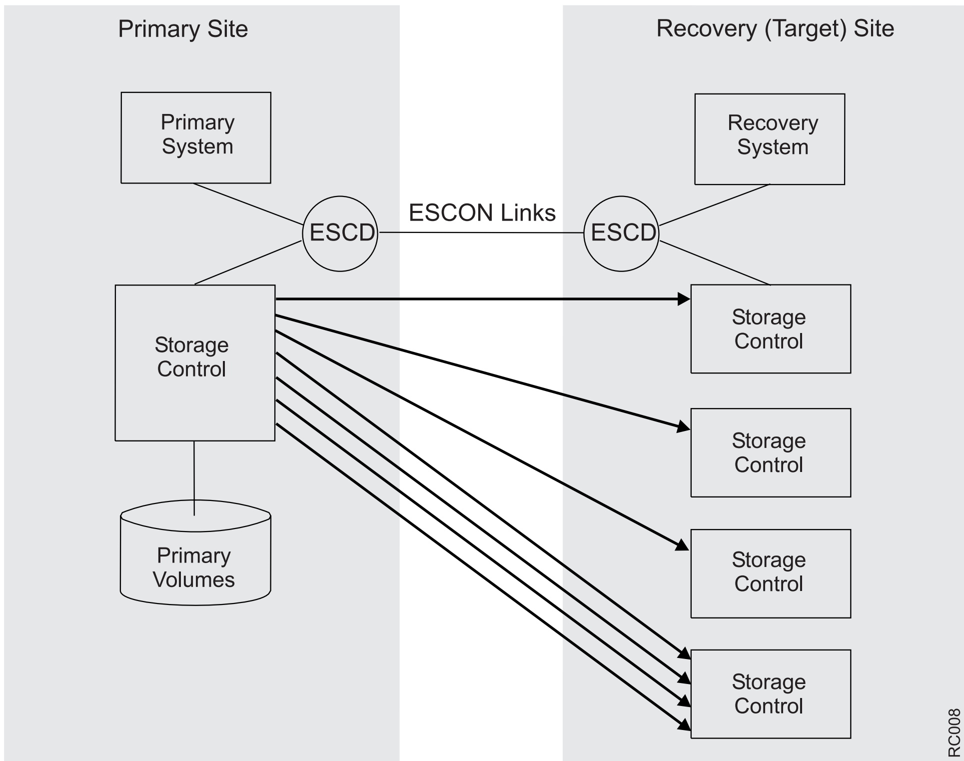


Figure 11. PPRC ESCON path options

A single storage subsystem at the recovery site can be linked to as many primary site logical subsystems as there are logical channels and devices available. A large cache and NVS help the recovery site logical subsystem to accept the copy workloads from multiple primary site logical subsystems.

You can use existing ESCON or FCP paths, or add other dedicated paths. ESCON paths can include an ESCON Director (ESCD) or Directors, or 9036 ESCON Remote Channel Extenders to provide configuration flexibility. FCP paths can be through fiber channel switch to provide configuration flexibility. PPRC over Fibre Channel allows a reduction in PPRC link infrastructures by a 4 to 1 ratio.

Note: You might have bandwidth restrictions using ESCON or FCP, if so, consider using the PPRC extended distance feature.“Managing PPRC extended distance mode” on page 353.

With FCP, a single fibre channel link between two ESS Model 800 ports enables bi-directional PPRC. This means that it is possible for data to travel in both directions simultaneously. For example, one link can have a PPRC path that is established in one direction and then have another PPRC path that is established in another direction at the same time, on the same physical path.

While a FCP interface can accommodate simultaneous data transfers, it does have limited bandwidth. To distribute workload evenly to all available paths, the ESS monitors the overall workload on each port and selects paths that are determined by the size of the data transfer, the available bandwidth available on each FCP port, and the number of data transfers, currently processing on each port. Selecting paths in this manner ensures good response time and overall system throughput.

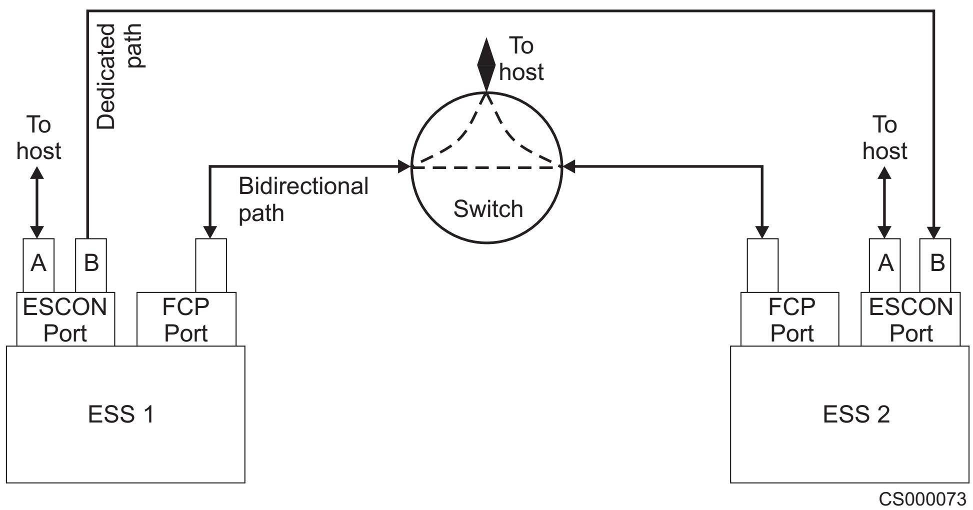


Figure 12. ESCON and FCP configuration

Considering the PPRC solution

Because of the haphazard nature of disasters, and the probability that during a disaster all of your data processing equipment will not fail at the same instant, IBM offers the following guidelines if you are considering PPRC as your disaster recovery solution:

- Choose PPRC and implement these system preparations:
  - Establish the links necessary to allow you access to primary system IEA49*xx* ERP messages from the recovery site. The IEA49*xx* message indicates that a PPRC volume pair has been suspended at the time indicated. Updates are written to the suspended primary volume after this suspension time but are not copied to the suspended secondary volume.

    Having a procedure in place to relay ERP messages to the recovery site is not, by itself, a 100% disaster recovery solution. In order to ensure data integrity for all PPRC secondary volumes, you must specify CRIT(YES) on *all* CESTPAIR commands .
  - Specify CRIT(YES) on CESTPAIR commands. The CRIT(YES) option ensures that writes are inhibited when the PPRC pair is suspended as a result of a failure between the primary and secondary storage controls. The PPRC copy operation remains suspended until the problem is corrected and either a CESTPAIR RESYNC or CDELPAIR command is issued.

    If the default setting is changed to CRIT(YES-ALL), then specifying CRIT(YES) on CESTPAIR commands means that future write operations are stopped as a result of any failure to update a secondary device. Failures here include primary and secondary device errors. Path errors and secondary storage control failures cause PPRC to suspend the pair and stop all writes to the primary volume.

    Note: The CRIT(YES) parameter has two modes of operation: CRIT(YES-PATHS) and CRIT(YES-ALL). The default setting, CRIT(YES-PATHS), inhibits all write operations only when a failure causes all paths to the secondary volume to be lost.

- Consider XRC instead of PPRC. The asynchronous solution, XRC, uses the system data mover, which manages the placement of data on the secondary volumes as well as recovery of those volumes.

For additional information about error recovery, see “Preparing for PPRC error recovery” on page 382.

Using PPRC with open system volumes

With Open System devices, you should consider the following as you plan your PPRC implementation:

Disk Considerations
: You must define at least one CKD device on the same cluster of the subsystem where the Open System device resides. This CKD device acts as an access device allowing you to issue the commands that control PPRC on the fixed block (FB) device. This means that if you want to manage both odd and even FB LSSs at least one odd and one even CKD LSS is required.

PPRC Path configuration
: There are two alternatives for configuring PPRC paths with Open LUN management. It is possible to either share the physical links between a CKD and an FB LSS or to use different physical links for the different environments.

If there is a requirement to be able to switch the Open LUN and System z environments independently, then a different set of physical links should be used for the two different sets of devices.

Note: This is not necessary if it is possible for paths to be defined in both directions on a single interface.

If the System z and Open LUN environments always have their primary devices in the same site, then it is possible to share the physical links, since there is no requirement to define paths in different directions. However, in order to avoid possible interference between the two environments you might choose to keep the links separate.

Determining which Open LUN volumes to designate for GDPS PPRC management
: Different open systems platforms might have different requirements for the volumes to be mirrored. These requirements do not change when implementing GDPS with Open LUN management. Consult the appropriate server and/or database platform documentation. It is recommended however, that all production volumes should be designated for PPRC use.

GDPS Disk naming conventions for Open LUN
: The Open LUN configuration requires that each ESS has a 4 character name assigned to it. Consult the *GDPS/PPRC INSTALLATION AND CUSTOMIZATION GUIDE* , ZG24–6703, for guidelines respecting this naming convention.

See Chapter 14, “Peer-to-Peer Remote copy command descriptions,” on page 291 to learn how to specify Open System devices.

Establishing the PPRC solution

The configuration of the PPRC system is vital to its recovery and performance capabilities. The following section describes basic PPRC configurations and those that include DASD channel extenders.

Synchronous PPRC solutions

Synchronous PPRC, also known as Metro Mirror, establishes a relationship between a primary source volume and a secondary copy volume in which both are updated simultaneously. The synchronous copy ensures that I/O completion of the application write to the primary is not signalled until after the write to the secondary is also complete. In a synchronous PPRC relationship, the secondary is always identical to the primary.

The advantage of a synchronous PPRC solution is that there is minimal host impact for performing the copy, since it is all handled by storage control hardware. No host resources are used to perform the initial copy or shadow subsequent updates. This is attractive to installations with limited CPU resources. The secondary storage control does not necessarily need to be channel attached to the application host, just the recovery host. This is attractive to installations with limited application host addressability.

The disadvantage is that since the copy operation is synchronous, there is an impact to application performance, as the application I/O operation will not be signalled as complete until the write to the secondary is also complete. The longer the distance between primary and secondary storage controllers, the greater this impact to application I/O, and therefore, application performance.

Plan your configuration to provide for capacity requirements, redundancy requirements, and performance requirements. Peer-to-peer remote copy allows from one to eight ESCON or FCP paths (IBM recommends at least two paths) from any primary site logical subsystem to a single recovery site logical subsystem. A specific primary site logical subsystem can connect with up to 16 different recovery site logical subsystems. However, you can link a single logical subsystem at the recovery site to a maximum of 64 primary site logical subsystems. A large cache and NVS will help the recovery site logical subsystem to accept the copy work loads from multiple primary site logical subsystems.

You can establish PPRC paths between the storage controls after you have installed the necessary Licensed Internal Code (LIC) in each storage control. You can use a mix of the following connections between your primary logical subsystem or storage control and the recovery logical subsystem or storage control:

- Dedicated ESCON channels
- Use existing ESCON channels and connect them using common ESCON Directors
- FCP channels
- Fibre channel switch option

When you are establishing paths, distribute them between different storage clusters (ESS) and Directors to gain flexibility and availability. If you have a path that involves an LSS to LSS pairing you can only have one type of path, ESCON or FCP. Paths for a source or target LSS can be mixed, as long as the different paths do not involve the same source and target LSSs.

Depending on the capability of your storage subsystems, you can establish up to eight paths with each CESTPATH TSO command. The following table indicates the information you need for establishing an ESCON path and a FCP path.

ESCON	FCP
Primary Storage Subsystem • Subsystem identifier (*ssid*) • Subsystem serial number (*serialno*) • LSS number is required if the storage control supports logical subsystems (like the ESS) and not allowed if the storage control does not support logical subsystems.	**Primary Storage Subsystem** • Subsystem identifier (*ssid*) • World Wide Node Name(*wwnn*) • LSS number is required if the storage control supports logical subsystems (like the ESS) and not allowed if the storage control does not support logical subsystems.
LINK • System Adapter ID interface identifier (*aaaa*) • Recovery site storage control ESCON link destination address (*bb*). The value is 00 for a direct ESCON connection, and contains the ESCON Director destination port if the connection goes through an ESCON Director. • Non-ESS devices the value is always 00 (*cc*). LSS devices require a value. See the documentation for the logical subsystem involved.	**LINK** • Fibre Channel adapter number (*aaaa*) in the primary storage subsystem • Fibre Channel adapter number (*bbbb*) in the secondary storage subsystem
Secondary (Recovery) Storage Subsystem • Subsystem identifier (*ssid*) • Subsystem serial number (*serialno*) • When ESS, Logical subsystem for the device (*lss*). LSS number is required if the storage control supports logical subsystems (like the ESS) and not allowed if the storage control does not support logical subsystems.	**Secondary (Recovery) Storage Subsystem** • Subsystem identifier (*ssid*) • World Wide Node Name(*wwnn*) • When ESS, Logical subsystem for the device (*lss*) LSS number is required if the storage control supports logical subsystems (like the ESS) and not allowed if the storage control does not support logical subsystems.

For additional information about the CESTPATH command, with its syntax and parameters, refer to "CESTPATH – establishing paths" on page 304.

For additional information and examples about establishing PPRC paths, refer to "Establishing PPRC paths" on page 325.

Asynchronous PPRC solutions

Global Copy

Global Copy, also known as PPRC-Extended Distance (PPRC-XD) , is a feature of the ESS allowing the primary and secondary storage sites to bridge great distances through channel extenders and telecommunication lines. In PPRC-XD, the primary updates the secondary incrementally, reducing impact to application writes and reducing bandwidth requirements. See "Managing PPRC extended distance mode" on page 353 for more information.

Global Mirror for ESS

Global Mirror combines PPRC-Extended Distance pairs (PPRC-XD) with FlashCopy consistency groups to provide true backup at a secondary site. This is accomplished through the addition of a master session in the hardware configuration. This master session controls all updates to the secondaries of the PPRC-XD pairs and, at the secondary site, the creation of consistent copies (using

FlashCopy), at user specified intervals. Global Mirror for ESS requires FCP connection. See Chapter 20, “Planning for Global Mirror,” on page 407 for more information.

Metro/Global Copy

Metro/Global Copy is a cascaded PPRC solution that enables you to combine two PPRC pairs, synchronous PPRC and PPRC-XD, which operates asynchronously, to create a low-cost, long-distance, tier four data loss disaster recovery configuration. In a cascading PPRC configuration, a single volume is defined as both a primary and a secondary, combining three volumes into two PPRC pairs.

In Figure 20 on page 289, we see two PPRC pairs, local pair A-B, and remote pair B-C. Volume B acts in two roles simultaneously. Volume B is the secondary volume of the local pair and the primary volume of the remote pair. As in any PPRC relationship, volumes A and B as well as volumes B and C must be in different logical subsystems (in the same or different ESS). There is a minimum of two different ESSs and a maximum of three different ESSs in a cascaded PPRC relationship. In most cases, volumes A, B, and C will be located in different sites: a local site, an intermediate, and a remote site.

“Setting up Metro/Global Copy” on page 289 explains how to configure Metro/Global Copy.

Metro/Global Mirror for ESS

Metro/Global Mirror adds FlashCopy relationships to insure consistent point-in-time data at the remote recovery site.

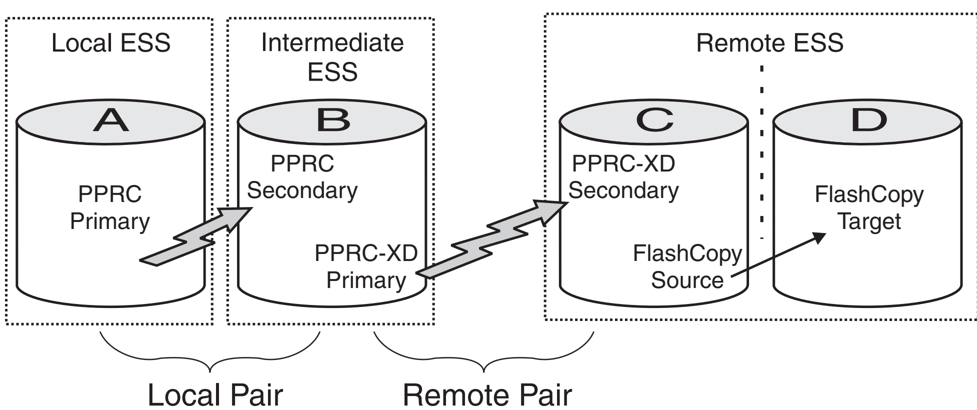


Figure 13. Metro/Global Copy with optional FlashCopy

Also known as Cascading PPRC with Global Mirror, Metro/Global Mirror is a chained set of PPRC pairs, the first pair being synchronous PPRC and the second pair being a PPRC-XD pair that is part of a Global Mirror session managed by a master storage control. The master storage controls FlashCopy consistency group formation across the session at regular intervals.

For information on converting an existing cascaded configuration to a multi-target configuration, refer to “Converting to Multi-Target Mirror” on page 275.

Multi-target PPRC solutions

With a Multi-target PPRC solution, a device can be the primary of more than one PPRC pair.

Multi-target PPRC allows you to combine copy services into many different configurations. The specific configurations that are available depend on the hardware. For information about the configurations that are available for your installation, refer to the representative and documentation for the hardware.

The topics that follow illustrate basic configurations that provide multi-target PPRC.

Multi-Target Metro Mirror

In a Multi-Target Metro Mirror configuration, the primary has two synchronous relationships.

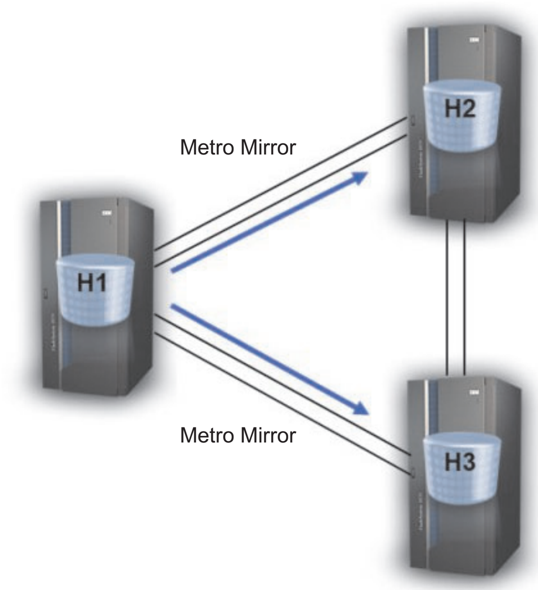


Figure 14. Multi-Target Metro Mirror

Data is transferred to both secondaries, H2 and H3, in parallel. The pairs operate independently of each other. They may be established, suspended or removed separately.

Multi-Target Metro/Global Mirror

In a Multi-Target Metro/Global Mirror configuration, the primary has a Metro Mirror relationship with one PPRC pair and a Global Mirror relationship with the other.

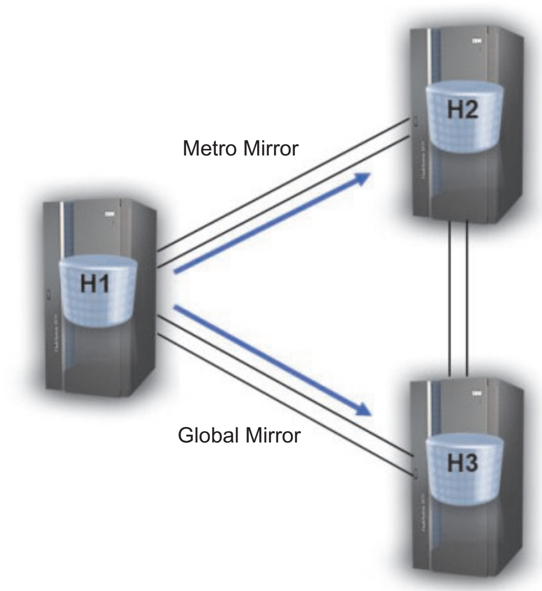


Figure 15. Multi-Target Metro/Global Mirror

The Metro Mirror pair is used for high availability in case of an outage that affects only the primary. The Global Mirror pair is used for remote disaster recovery.

Multi-Target with Global Mirror and Global Copy

In this configuration, there are two asynchronous relationships, a Global Mirror and a Global Copy.

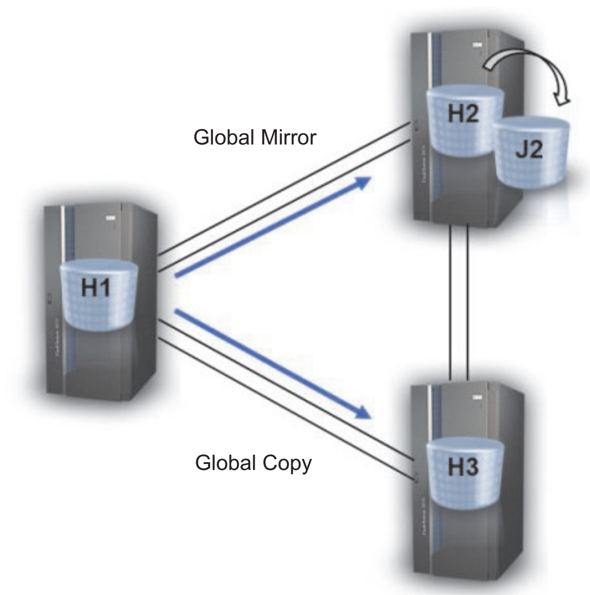


Figure 16. Multi-Target with Global Mirror and Global Copy

This configuration could be used for migration purposes or to provide a resilient long distance disaster recovery solution between three locations where it is possible to restart or retain a Global Mirror distance disaster solution with the loss of any of the three locations.

Multi-Target with two Global Copy pairs

In this configuration, there are two Global Copy relationships.

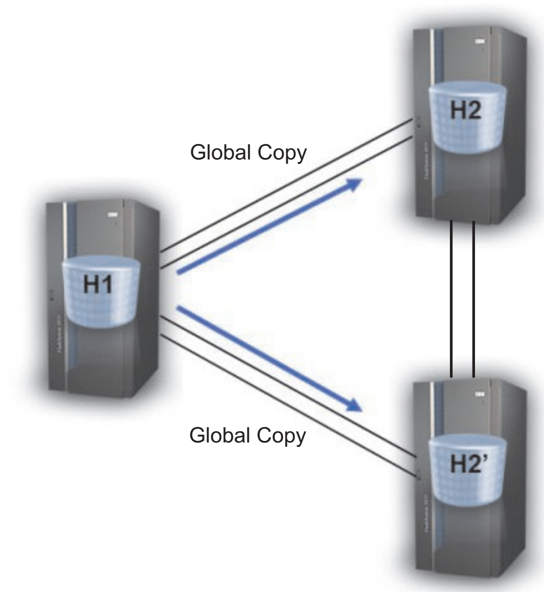


Figure 17. Multi-Target with two Global Copy pairs

This configuration is a transition state for various configurations where Global Mirror is in use or where Global Copy is used to provide a reduced impact resynchronization procedure.

Internal pairs in a Multi-Target Mirror configuration

Once a multi-target configuration is created, an incremental relationship is created internally between the secondary devices.

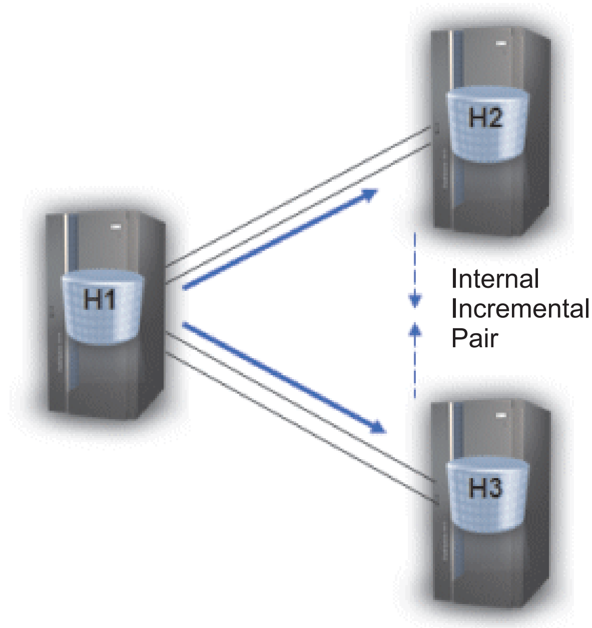


Figure 18. Internal Incremental Pairs in a Multi-Target Configuration

The incremental relationship can minimize the amount of data that must be copied to resynchronize the new pair in a recovery scenario, because only the changed data is copied. To use the internal incremental relationship to resynchronize the secondaries, you must establish paths between them. Then, you can resynchronize them with the PPRC FAILBACK command. It is recommended, but not required, that you establish the paths between the secondaries before creating the multi-target configuration.

If there are any problems establishing the internal relationship, the system retries until it is successful.

If a pair is removed from the multi-target configuration, the internal pair is also removed.

Converting to Multi-Target Mirror

To convert an existing cascaded configuration to a Multi-Target Mirror configuration, you perform an establish pair request with the MTFAILOVER parameter. For example, in a cascaded configuration, H1->H2->H3, where H1 is the primary, you issue a CESTPAIR command with ACTION(MTFAILOVER) to the middle device, H2, as shown in Figure 19 on page 276.

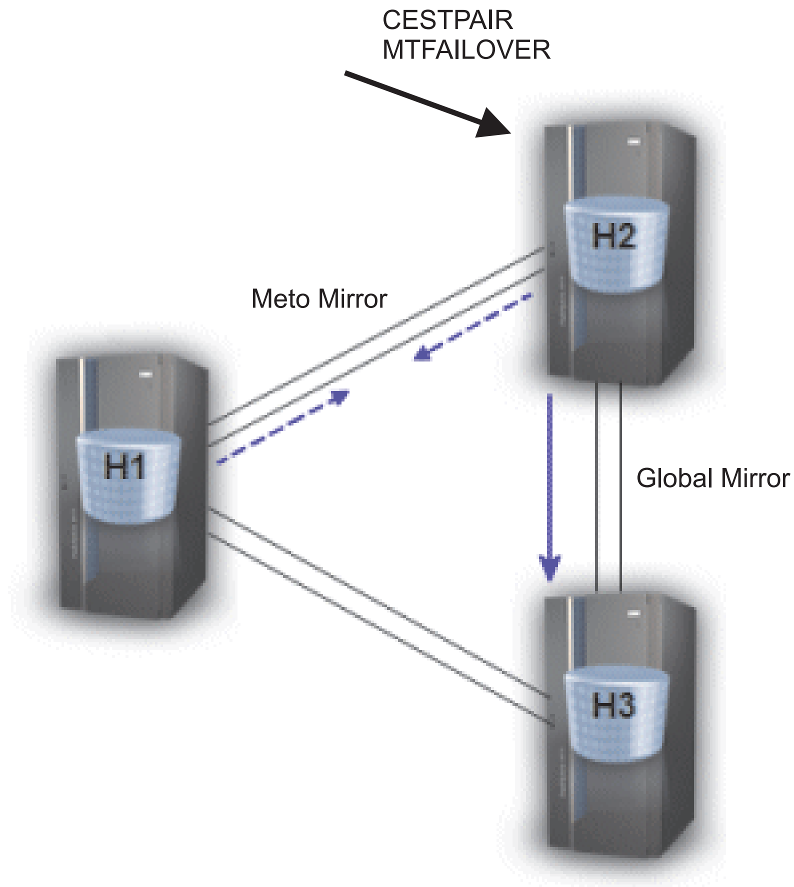


Figure 19. Initiating Failover H2 to H1 with MTFAILOVER

This results in a multi-target configuration (H2->H1 and H2-H3) in place of the previous cascaded failover state

PPRC and DASD channel extenders

Adding channel extenders to a synchronous PPRC configuration to support longer distances can seriously degrade the application performance. This is because of the additional time that is added to each copy write I/O. IBM therefore does not generally recommend using channel extenders with PPRC in synchronous mode for any applications that are sensitive to response time. See “Using channel extenders” on page 20 for general information about channel extenders.

However, the PPRC does offer a long-distance solution with little application response degradation, even when using channel extenders. For more information, see “Managing PPRC extended distance mode” on page 353.

Determining PPRC resource needs

This section contains information and recommendations about the resources needed to establish a PPRC configuration that meets disaster recovery and workload migration needs. Evaluate the resource issues listed in Table 37 on page 277 before installing PPRC.

Table 37. PPRC resource considerations

Topic . . .
"Steps for evaluating how PPRC affects the primary storage subsystem"
"Evaluating how PPRC affects the recovery storage subsystem"
"Determining the recovery site storage capacity" on page 278
"Determining recovery system access to resources" on page 278
"Establishing storage subsystem channel connections" on page 278
"Establishing PPRC ESCON connections" on page 278
"Establishing PPRC FCP connections" on page 279

Steps for evaluating how PPRC affects the primary storage subsystem

Storage subsystem use can affect overall system performance, and it is therefore necessary to determine how adding PPRC might affect the primary storage subsystem.

Perform the following steps when evaluating how PPRC affects the primary storage subsystem:

1. Determine the write rate, for each primary site storage control, for the volumes that you plan to copy.
2. Add this rate, as if it was all DASD fast write, to the total storage subsystem use.
3. Decide if this impact on the storage subsystem allows applications to achieve acceptable disk I/O performance. If it does not, you might need to redistribute volumes among storage controls or add an additional primary site storage control.

Also consider how initial volume copy and resynchronization activity affect storage control use and application performance. You might notice this when you start a copy operation, restart a system, perform some copy recovery events, or when you add volumes to the PPRC system.

If you have a storage control configured with fixed block devices and LSSs, contact your DASD vendor to understand whether your primary and secondary disk subsystem support the prerequisite level of the PPRC architecture.

Evaluating how PPRC affects the recovery storage subsystem

PPRC lets you allocate recovery system volumes on any number of recovery storage subsystems. The storage controls at the recovery site do not have to have a one-to-one correlation with their primary storage subsystem counterparts. Project the performance impact on recovery storage subsystems to be sure that it is within acceptable limits.

Higher recovery storage subsystem activity increases the write time between storage controls, thereby affecting primary disk I/O performance. Include all other recovery storage subsystem activity in your assessment, such as access by other applications and systems to non-PPRC volumes on the same storage subsystem.

Determining the recovery site storage capacity

As part of planning for PPRC installation, you must consider the number and type of disk volumes. Use Table 38 to determine your necessary disk capacity.

Table 38. Steps for disk volume determination

Stage . . .	Description . . .
Determine the number of necessary recovery storage subsystem disk volumes.	Assume that all volumes from the primary subsystem must be copied to the recovery system. You can then consider each volume individually to determine if you need it for recovery and subsequent application operations on the recovery system. **Example:** You do not have to copy disk volumes that contain only page data sets to the recovery site storage subsystem.
Determine the number of disk volumes to be copied to the recovery site storage subsystem.	Consider which application and support data sets will be needed to start applications on the recovery subsystem at recovery time. **Example:** User catalogs are a good example of such data sets. If the recovery subsystem cannot find data volumes associated with user catalogs, jobs will not start successfully after recovery.
Consider disk volumes	Consider what disk volumes, such as DFSMShsm migration level 1 data sets, may be needed for recovery operations.

Determining recovery system access to resources

Rules: To meet disaster recovery needs, PPRC recovery systems must:

- Be able to run the applications that use the copied data.
- Be able to process commands to the secondary disk to end the secondary copy state and to become available for application system allocation.
- Have access to the TSO command module.

Establishing storage subsystem channel connections

Disk channel connections include those to primary and recovery systems. PPRC connections between storage subsystems can be ESCON or FCP connections. The use of fibre channel allows a reduction in PPRC infrastructures by a 4 to 1 ratio while delivering equivalent or better performance.

Tip: It might be useful when planning an initial PPRC test installation (with a limited number of volumes) to have both the primary and secondary volumes on the same storage subsystem, thereby conserving resources.

Establishing PPRC ESCON connections

ESS storage subsystems support synchronous PPRC at a distance of up to 103 km. You can configure other IBM storage subsystems to support ESCON ports (connected by multimode fiber optic cables) at distances of up to three kilometers. ESCON Directors, if your configuration includes them, can have Extended Distance Feature (XDF) ports with singlemode fiber-optic cables for longer link lengths.

Note: PPRC allows data to be mirrored to distances longer than 103 km, using channel extenders. For information, see "Managing PPRC extended distance mode" on page 353.

You can use 9036 ESCON Remote Channel Extenders to convert from multimode fiber links to XDF and back, and as XDF extender links. Use these or other comparable repeaters when the distance between systems is greater than 3 km and your configuration does not include ESCON Directors.

You can only dynamically switch one ESCON Director within a path to an I/O unit. The other ESCON Director, if present, provides a static connection. Refer to the appropriate storage subsystem planning guide for information about ESCON configurations with ESCON Directors.

Recovery site storage subsystems must have access to the host systems that will run primary applications when the primary system is out of service. A host system at the recovery site must be able to issue recovery commands as part of recovery system takeover procedures.

Establishing PPRC FCP connections

There are two differences between establishing a PPRC FCP path versus a PPRC ESCON path:

- You must specify the World Wide Node Name for the primary logical subsystem and the secondary (recovery) logical subsystem instead of a serial number.
- You must specify the FCP source and destination adapter numbers in the LINK addresses.

Each ESS has a unique World Wide Node Name (WWNN). The WWNN is displayed on the initial panel of the ESS Specialist. The WWNN is an 8 byte value. The WWNN for the primary logical subsystem is checked to ensure the primary WWNN specified in the CESTPATH TSO command or the PESTPATH ANTRQST API request is valid.

To establish an FCP link between subsystems, the establish path request must supply a WWNN for the primary logical subsystem and a WWNN for the secondary logical subsystem. You cannot specify a WWNN for one logical subsystem and a serial number for the other logical subsystem, this causes the establish path command or request to fail.

When an FCP connection between ESS subsystems is required, the LINK address(es) specify an FCP adapter in the primary logical subsystem, and an FCP adapter in the secondary logical subsystem. The FCP adapters are assigned numbers during ESS configuration. The FCP adapter numbers are two bytes in length. Although the first byte of the FCP adapter number is generally zero, two bytes are needed for future FCP adapter configurations. When the PPRC establish paths command or request parameter checking is successful, the WWNN of the secondary storage subsystem and the FCP adapter number are combined to create the World Wide Port Name (WWPN) address. The WWPN is sent directly to the secondary subsystem (direct point-to-point connection) or is sent to the Fibre Channel fabric switch where the request is routed to the appropriate secondary subsystem. If a secondary subsystem with the requested WWPN is found, the secondary subsystem FCP adapter assigns a port in the adapter to receive PPRC communications from the primary subsystem. Although the FCP adapter port in the secondary subsystem is linked to the primary subsystem, the secondary FCP adapter port can be used for other communications as needed.

When the PPRC link between subsystems is no longer needed, you must issue a CDELPATH TSO command or PDELPATH ANTRQST API request to remove all links between the primary storage subsystem and the secondary storage subsystem. Deleting the paths allows the use of the affected storage adapters (ESCON or FCP) for other communications. Before the you delete all the paths, all PPRC volume pairs must be deleted.

Synchronous PPRC is supported over FCP for distances up to 200 km.

Note: If the paths were established using WWNN for the primary and secondary, the CDELPATH must also specify WWNN for the primary and secondary.

For additional information about using the CESTPATH AND CDELPATH TSO commands, refer to Chapter 14, "Peer-to-Peer Remote copy command descriptions," on page 291.

For additional information about using the PESTPATH and PDELPATH ANTRQST API requests, refer to Appendix C, "ANTRQST and ANTRQSTL macros – call to the system data mover API," on page 573.

Copying the catalog and control data sets

You can reduce disaster recovery time if you use remote copy for the volumes that contain the master catalog, key user catalogs, and system control data sets. This may be a better choice than maintaining a current copy of the catalog and various control data sets on the recovery system.

Remote copy automatically copies all data sets on the managed volumes. These include catalog data sets and control data sets.

Controlling access to PPRC commands

The following are two ways to protect access to PPRC commands. You can:

- Put the commands in a RACF-protected library.
- Define resource profiles in the RACF facility class and restrict access to those profiles.

For additional information about putting commands in a RACF-protected library, refer to "Steps for placing PPRC commands in a RACF-protected library" on page 282.

For additional information about how to define resource profiles in the RACF facility class, refer to "Defining resource profiles in the RACF Facility class" on page 282.

Chapter 13. Setting up the Peer-to-Peer Remote copy environment

This chapter describes how to set up the peer-to-peer remote copy (PPRC) environment.

In this topic

The following sections are included in this chapter:

Installing Peer-to-Peer Remote copy

To install peer-to-peer remote copy, perform the following steps, after installing the appropriate z/OS release:

1. Authorize PPRC TSO commands by adding the command names to the AUTHCMD PARM parameter of the IKJTSO*xx* member of SYS1.PARMLIB. After you have added the PPRC command names to the IKJTSO*xx* member, issue the TSO command PARMLIB UPDATE(*xx*) to activate the new IKJTSO*xx* member.

   To restrict PPRC commands to certain users, follow the instructions in "Controlling access to PPRC resources."
2. Based on the applications that you plan to copy, determine which devices you want to establish as primary and secondary devices.
3. Determine the number of paths that are required to establish connections between each primary site and recovery site logical subsystems. This number is based on the number of primary and secondary devices involved.
4. Use the DEVSERV system console command to determine the channel connection addresses for the devices. The serial number is on the lower left-hand part of the operator's panel on an IBM storage control.
5. Issue the appropriate CESTPATH TSO commands to establish the PPRC paths.
6. Issue the appropriate CESTPAIR TSO commands to define the PPRC pairs.
7. Issue the TSO CQUERY PATHS and CQUERY VOLUME commands to verify the PPRC paths and pairs configuration.

Controlling access to PPRC resources

Each installation is responsible for preventing the unauthorized use of PPRC TSO commands and for controlling ICKDSF activity to PPRC-managed volumes. This section describes how to protect PPRC commands and control volume repair activities.

Protecting access to PPRC commands

If possible, restrict PPRC TSO commands to authorized storage administrators only. You can place the commands in a library that is protected by the IBM Resource Access Control Facility (RACF), and define access to resource profiles.

Steps for placing PPRC commands in a RACF-protected library

Place the TSO commands in a RACF-protected library to restrict PPRC TSO commands to authorized storage administrators.

Before you begin: You need to be familiar with the commands used in these steps.

Perform the following steps to RACF-protect PPRC commands:

1. Issue the following RDEFINE command for each PPRC command, and for each command abbreviation that you want defined to RACF:

   ```
   RDEFINE PROGRAM cmdname ADDMEM('SYS1.CMDLIB')/volser/NOPADCHK) UACC(NONE)
   ```

   The following terms apply to the above example:

   cmdname
   : Defines the PPRC TSO command name or an abbreviation of a command. Issue a separate RDEFINE command for each command and any command abbreviations you plan to use. RACF can only perform checking on commands and abbreviations that are defined to it.

   volser
   : Defines the name of the volume that contains the SYS1.CMDLIB data set.

2. Issue the PERMIT command for all commands and authorized PPRC TSO command users as follows:

   ```
   PERMIT cmdname CLASS(PROGRAM) ID(name) ACCESS(READ)
   ```

   The following terms apply to the above example:

   cmdname
   : Defines the PPRC TSO command name, or an abbreviation of a command.

   name
   : Defines the user ID that will receive RACF access authority for that command name.

3. Issue the SETROPTS command from a user ID that has the appropriate authority:

   ```
   SETROPTS CLASSACT(PROGRAM) WHEN(PROGRAM) REFRESH
   ```

For additional information about the peer-to-peer remote copy commands, refer to Chapter 14, “Peer-to-Peer Remote copy command descriptions,” on page 291.

Defining resource profiles in the RACF Facility class

You can limit the use of PPRC commands by defining resource profiles in the RACF FACILITY class and restricting access to those profiles. To use a protected command, you need read-access authority to the applicable profile.

Table 39 on page 283 lists the PPRC commands and the facility class profiles that can restrict them. See the *z/OS Security Server RACF Security Administrator's Guide* for details on activating the RACF facility class, and defining and authorizing users to the PPRC command profiles.

Table 39. PPRC FACILITY class profile names

Command	Profile Name
CDELPAIR	STGADMIN.ANT.PPRC.COMMANDS
CDELPATH	
CESTPAIR	
CESTPATH	
CGROUP	
CQUERY	
CRECOVER	
CSUSPEND	
PSETCHAR	
CQUERY	STGADMIN.ANT.PPRC.CQUERY

Note: Authorize CQUERY command use with the STGADMIN.ANT.PPRC.COMMANDS profile or the STGADMIN.ANT.PPRC.CQUERY profile. PPRC first checks STGADMIN.ANT.PPRC.COMMANDS for authorization. If authorization is not permitted with the STGADMIN.ANT.PPRC.COMMANDS profile, PPRC checks the STGADMIN.ANT.PPRC.CQUERY profile for authorization to issue the CQUERY command.

Examples: The following examples activate the RACF FACILITY class, define the profile for the PPRC commands, and give user STGADMIN authority to use this profile:

- This example activates the RACF FACILITY class:

```
SETROPTS CLASSACT(FACILITY)
```

- This example defines the profile for PPRC commands, and authorizes user STGADMIN to use this profile:

```
RDEFINE FACILITY STGADMIN.ANT.PPRC.COMMANDS UACC(NONE)

  PERMIT STGADMIN.ANT.PPRC.COMMANDS CLASS(FACILITY) -
         ID(STGADMIN) ACCESS(READ)
```

Controlling ICKDSF activity to PPRC volumes

The storage administrator must control ICKDSF activity to primary and secondary PPRC-managed volumes. PPRC cannot copy some ICKDSF operations from the primary to the secondary volumes. If you plan to use ICKDSF to repair either a primary or secondary volume, first remove the volume pair from the session. After the ICKDSF function is complete, reestablish the session with a CESTPAIR command.

Performing an ICKDSF repair is described in “Steps to perform an ICKDSF repair on a PPRC-managed primary volume:” on page 284 and “Steps to perform an ICKDSF repair on a PPRC-managed secondary volume:” on page 284.

Attention: Failure to follow the procedures of an ICKDSF repair can result in potential data integrity exposures to offline volumes.

If you are planning to repair or replace a primary volume, you might want to use the P/DAS function to minimize impact to the application I/O while the error on the primary volume is being corrected.

For additional information about specific error conditions, refer to "Putting PPRC error recovery procedures into effect" on page 382.

For additional information about the P/DAS function, refer to Chapter 19, "Peer-to-Peer Remote Copy dynamic address switching (P/DAS)," on page 389.

Steps to perform an ICKDSF repair on a PPRC-managed primary volume:

Use the following steps to perform an ICKDSF repair on a PPRC-managed *primary* volume:

1. Suspend the volume pair with a CSUSPEND PRIMARY command that is directed to the primary volume. This assumes that the volume pairs are not already in a suspended state as a result of the error that necessitated the ICKDSF repair action.
2. Run the latest level of ICKDSF to correct the error once the pair is in the suspended state.
3. Remove the volumes from the suspended state by using the CESTPAIR command with the RESYNC option to resume operations.

For additional information about specific error conditions, refer to "Putting PPRC error recovery procedures into effect" on page 382.

Steps to perform an ICKDSF repair on a PPRC-managed secondary volume:

Use the following steps to perform an ICKDSF repair on a PPRC-managed *secondary* volume:

1. Issue a CESTPAIR command with ACTION(FAILOVER). This puts the secondary in a primary suspended state.
2. Run ICKDSF release 17 or above to repair the volume.
3. Issue a CESTPAIR command with ACTION(FAILBACK).

Identifying volume pairs

The storage administrator must identify application volumes to copy, and create peer-to-peer remote copy volume pairs. PPRC provides an image copy of a volume on a record-for-record basis. There is a one-to-one correspondence between the record on the primary volume track and the record on the secondary volume track. PPRC writes data on the same tracks on the secondary disk as it does on the primary disk. Therefore, the secondary disk must have the same track sizes and number of tracks per cylinder, and either the same or larger volume capacities, as the primary. If the disk does not meet these criteria, the CESTPAIR command will fail.

You should make secondary volumes part of a storage group that does not allow allocations, and dedicate these volumes to PPRC use only. PPRC secondary volumes are similar to 3390 dual copy secondary volumes. All host I/O operations directed to these offline secondary volumes are command rejected.

Remote copy supports volumes, not data sets. As a result, all data sets on the volumes that are copied are part of remote copy activity, and are therefore copied to the recovery system. This support is application (IBM or non-IBM) independent, and supports all data set types.

Because applications deal with data sets and not volumes, multivolume data sets require special attention. Unless you copy all volumes of a multivolume data set, PPRC will only copy part of the data set. Other data sets on the copied volume may be usable, but the multivolume data sets will not be. If multivolume data sets are critical for recovery, you must also copy the other volumes on which these data sets reside.

Note: Multivolume data set types include data sets that reside on multiple volumes, striped data sets, and VSAM spheres.

Carefully choose the volumes to copy so as not to unnecessarily affect overall PPRC performance. Copying page data sets can affect performance.

It would be wasteful to have PPRC copy a volume that contains page or other data sets that are specific to the application host system.

For additional information about the CESTPAIR command, see "CESTPAIR – establishing volume pairs" on page 296.

Steps for including PPRC TSO commands in automated procedures

You can use PPRC TSO commands in automated procedures to do certain PPRC functions. The description below, for example, outlines an automated procedure to check PPRC volume status at system IPL time and resynchronize volumes that are in suspended or duplex state.

Before you begin: You need to be familiar with the commands used in these steps. For more information, refer to Chapter 14, "Peer-to-Peer Remote copy command descriptions," on page 291.

Perform the following steps to include PPRC TSO commands in automated procedures:

1. Put the CQUERY PPRC TSO command in either a batch JCL job or PROCLIB member to query the status of all primary volumes.
2. Automatically start the procedure at IPL time by putting the procedure name into SYS1.PARMLIB(COMMND*xx*). PPRC sends the results from the query to the system's log.
3. Use NetView, or other console automation product, to capture the output from the CQUERY command and examine specific path and volume status information.
4. Use NetView, or another console automation product, to initiate jobs for any paths and volume pairs that you need to reestablish. These jobs can be other procedures and can use JCL substitution.

Note:

1. Consider attaching a host to the secondary subsystem. If regular production operations are going to take place at the secondary subsystem, install an ECAM device there to monitor real space usage at that secondary LSS. An ECAM device cannot be a secondary volume of a PPRC pair, however, because the secondary device cannot receive I/O.

2. Some storage subsystems only send the SIM down the path that issues the next SIO. Many subsystems, however, send the message down every active path group. As a result, any host that is attached to the subsystem can receive the ERP SIM messages. This is an important consideration as you plan for disaster recovery.

Configuring storage subsystem resources for PPRC

All synchronous copy operations, including PPRC, have an inherent performance penalty: Write operations to the primary subsystem must wait until the data is secure in the secondary subsystem before sending completion status. This necessary step adds a minimum of an additional fast write operation plus the data transmission times.

You must properly configure your PPRC resources to do the following:
- Minimize additional impact on system performance.
- Avoid unintended interruptions of the remote copy function.
- Maximize the data integrity of the secondary volumes.

Note: PPRC extended distance, an asynchronous mode of operation, allows primary systems to complete application I/O before sending it to the secondary volume.

For additional information about PPRC extended distance, refer to "Managing PPRC extended distance mode" on page 353.

Configuring IBM storage controls

Ensure that the service information message (SIM) alert reporting is correctly set at both PPRC storage subsystems.

The following are additional concerns for the storage controls at the recovery site:
- Attachment to at least one host for SIM reporting
- Use of remote power sequence control cables

Storage subsystem settings for SIM alerts

Set up all storage controls with PPRC operations so that PPRC routes SIM alert messages to a host console. Do not select "No Alerts", as this option prohibits the storage control from sending any SIM alert messages to attached host systems. In the event of a subsystem failure that affects PPRC operation (such as a permanent error on a secondary volume), the subsystem must be able to alert an operator to the problem. This is especially critical at an unattended recovery site where a subsystem problem, if left unresolved, can jeopardize the entire PPRC copy operation.

Attaching a recovery site storage control to a host system

It is recommended that you keep at least one host system attached to each recovery site storage control. You can locate the host at either site, as long as it can relay SIM alert messages to a console. This is necessary so that the storage controls for the secondary volumes can offload important SIMs in the event of a problem on the recovery storage subsystem.

Using remote power sequence control cables

Make every effort to minimize power interruptions to PPRC recovery site storage subsystems.

Recommendation: It is recommended that you do not use power sequence control cables between host processors and attached storage controls that contain PPRC secondary volumes. These cables, when installed, allow an attached host processor to remotely remove power from the storage controls. If power sequence control cables are present, ensure that the power-select switches for both clusters are in the LOCAL position. The power-select switches are on an IBM Storage Control's power sequence control board. This setting removes the possibility that a processor at the recovery site can inadvertently power down a PPRC recovery storage subsystem.

Accessing PPRC secondary volumes

Dedicate all PPRC secondary volumes exclusively to PPRC use. The system automatically rejects all host read and write access as long as the PPRC pair remains active in the session, regardless of whether the secondary volume is online or offline.

Accessing open system devices

Open System (fixed block) devices can be included in PPRC relationships provided that a CKD device is located in the same subsystem cluster as the fixed block device. This CKD device acts as an access device allowing you to issue the commands that control PPRC on the fixed block (FB) device. See Chapter 14, "Peer-to-Peer Remote copy command descriptions," on page 291 to learn how to specify Open System devices.

Optimizing PPRC performance

This section describes how synchronous operations (like peer-to-peer remote copy) can affect various work loads, and contains guidelines for getting the best PPRC performance. See your IBM marketing representative for the latest remote copy performance and configuration information and recommendations for your specific environment.

Note: PPRC extended distance, an asynchronous mode of operation, might offer additional optimization considerations beyond those available for synchronous PPRC.

For additional information about PPRC extended distance, refer to "Managing PPRC extended distance mode" on page 353.

Analyzing work load characteristics

For synchronous operations there is additional overhead for every write. Several factors can affect the overall performance of write operations, especially for synchronous copy operations. The following work load characteristics can affect synchronous write performance:

- Write content — There is an inherent performance penalty on synchronous operations that are associated with write operations. As a result, work loads with high read-to-write ratios experience much less impact than those that are write intensive. As more writes occur to the primary site storage control, there is a corresponding effect on performance.

  Some examples of extremely write-intensive work loads are SPOOL volumes and database logging volumes, which can affect I/O response times dramatically in any synchronous copy operation. A possible solution to this high write-intensive work load is to spread the SPOOL volumes across multiple volumes to minimize queueing.
- Blocksize — Work loads with small blocksizes experience better performance than do work loads with large blocksizes. As the write blocksize increases, the

affect on work load performance also increases. One example of a work load that has this impact is DB2® deferred writes. A possible solution is to spread the work load across several volumes.

- Overall channel demand — As the demand increases on the host channel that services the primary application's storage control, the impact to overall performance also increases. Spread the work load across several storage controls if possible.
- Sequential write operations — Batch processing operations typically exhibit high sequential write operations. Synchronous copy operations can result in longer batch windows.

  Evaluate dump restore activities to PPRC-managed volumes. PPRC uses a data flow algorithm to copy data from primary volumes to secondary volumes during initialization. To best optimize this algorithm, avoid restoring data sets or volumes to PPRC-managed volumes. Although PPRC supports restore operations, doing so increases the amount of data that PPRC must copy from the primary to the secondary volumes. You may choose to allocate a scratch volume on the primary system, and then restore data to that volume. Establish the PPRC session, and allow normal primary-to-secondary volume processing to continue.

Balancing storage control configurations

Configure your system so that primary system activity does not exceed the capacity of the recovery site logical subsystem. You might require performance at the secondary site that is equivalent to your primary application's performance. This could be performance when migrating work loads, or performance during a disaster recovery. If so, it is recommended that the secondary site logical subsystems have at least the same cache and NVS capacity as the primary site logical subsystems.

PPRC supports configurations where several primary site logical subsystems can funnel their updated data to a single logical subsystem on the recovery system. The number of copied primary volumes and the rate of write activity to these volumes can affect performance.

Ensure that you have a sufficient number of ESCON or FCP paths established between the primary and secondary site logical subsystems. This is especially important in configurations where the same logical subsystems manage both primary and secondary volumes.

Distributing work loads

Avoid directing all update activity to a small set of common volumes on a single recovery site storage control. The performance impact at the recovery site storage control adversely affects the primary site performance.

Typically, only a few devices on a particular storage control receive most of the activity. Usually two to four devices account for 50% or more of the total update activity. If, for example, two primary storage controls channel their most active volumes into a single recovery site storage control, the recovery subsystem could become overcommitted with remote copy activity.

Table 40 provides a useful summary of PPRC session configurations.

Table 40. Quick reference for PPRC session limits

You can configure up to . . .		
16 PPRC recovery site storage subsystems	for each	PPRC primary site storage subsystem

Table 40. Quick reference for PPRC session limits (continued)

You can configure up to . . .		
8 PPRC paths	between	primary and secondary LSSes

Using FlashCopy target as PPRC primary

This FlashCopy option allows you to establish a source and target relationship in which the FlashCopy target is the primary device in a PPRC pair. In this way you can create a point-in-time copy and the duplicate that point-in-time copy at a remote site. See "FlashCopy to PPRC primary" on page 486 for more information.

Setting up Metro/Global Copy

Metro/Global Copy relationships are established, suspended, and terminated for each pair separately. You should first establish the remote B-C pair, specifying the CASCADE (YES) option and the MODE (NOCOPY) options. It is recommended that CGROUP(YES) not be specified when establishing this path, since consistency groups are not applied to remote relationships. Then establish the local A-B pair, specifying the OPTION (SYNC) and MODE (COPY) options. Other options depend on your specific environment.

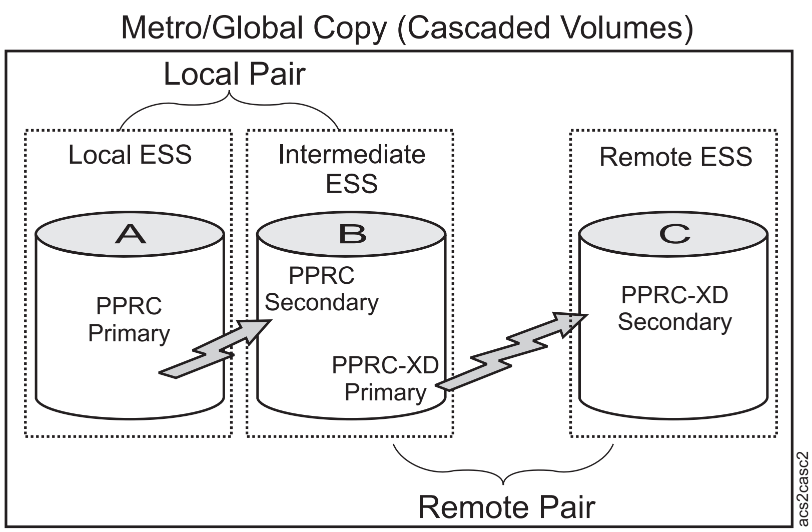


Figure 20. Metro/Global Copy (Cascaded Volumes)

In order to ensure a tier four data loss solution, periodic synchronization of the B-C pair is required to complete all updates to the remote secondary. This is accomplished by freezing updates at volume A and determining that volume C is in a consistent state with that point in time, either by waiting for all updates to be copied from volume B, or by converting the PPRC-XD pair to a synchronous relationship so that all pending updates are processed sequentially. (Synchronizing the PPRC-XD pair can have a significant impact on performance and should be scheduled with care.) When all updates to volume C have completed, make a FlashCopy or tape backup, resume I/O operations at the local pair and restore the remote pair to its asynchronous state. "Scenarios using PPRC extended distance and synchronous mode for backup purposes" on page 358 explains this procedure in greater detail.

Metro/Global copy mode combinations

Table 41 shows which mode combinations are supported in a cascading relationship.

Table 41. Supported Metro/Global Copy Mode Combinations

Local Volume Pair	Remote Volume Pair
Synchronous PPRC	Synchronous PPRC
	PPRC Extended Distance
PPRC Extended Distance	PPRC Extended Distance

The combination of PPRC-XD to synchronous PPRC is not supported because the intermediate volume B, a fuzzy copy, cannot be the source for a consistent copy at the tertiary volume.

Synchronous PPRC to PPRC-XD is the typical configuration intended for disaster recovery. The combination of PPRC-XD to PPRC-XD allows you to maintain multiple copies of data without signification impact to application I/O performance. In contrast, a combination of two synchronous PPRC relationships will cause a suspend on the first I/O to the first PPRC primary in the cascaded triad. This configuration may be useful to guarantee a consistent copy at the tertiary volume if there is little write I/O and the whole distance exceeds the maximum of 103 km supported for a single synchronous PPRC.

While following the rule that prohibits the two volumes that form a PPRC pair from being in the same LSS, it is theoretically possible to have these logical subsystems in the same ESS, however it is only practical to cascade PPRC pairs two ways:

- Three ESSs with one volume per ESS, as shown in Figure 13 on page 271

  When the local pair runs in synchronous mode, keep the distance between the local and intermediate ESSs as close as possible to minimize the performance impact. The intermediate ESS should be in a secure environment, separated from the local ESS, with separate power to reduce the possibility of an outage affecting both locations.

  The remote ESS is typically in a difference city, any distance from the intermediate ESS. If the remote PPRC pair operates asynchronously from the local PPRC pair, the effect of long-distance on host response time is minimized.
- Two ESSs with volumes A and B in the same local ESS and volume C in a remote ESS

  If the remote pair operates in PPRC-XD mode, this configuration is similar to a single PPRC-XD relationship. The intermediate volume allows procedures for consistent, remote, point in time copy with less application impact.

Chapter 21, “Setting up Global Mirror for ESS,” on page 411 describes how to set up Global Mirror.

“Setting up a Metro/Global Mirror configuration” on page 452 describes how to set up Metro/Global Mirror.

Chapter 14. Peer-to-Peer Remote copy command descriptions

This section describes peer-to-peer remote copy TSO commands with their required and optional parameters, and includes an example for each command. Chapter 15, "Managing Peer-to-Peer Remote Copy operations," on page 321 contains other examples.

In this topic

The following subjects are included in this topic:

PPRC command overview

Peer-to-peer remote copy TSO commands have the same restrictions as other TSO commands. The TSO abbreviation convention requires that you specify as much of the command parameter as is necessary to distinguish it from other parameters. Do not abbreviate PPRC command names.

PPRC TSO commands, like most TSO commands, are deferred requests. You receive an initial message that reports whether the command syntax is valid, and whether TSO has accepted the command for processing. Command function processing is a separate operation that can take minutes or hours to complete. The TSO terminal, meanwhile, is free to do other processing.

When TSO is operational, you can issue the PPRC commands to automatically initiate PPRC subsystem activity. Table 42 shows which PPRC TSO commands you can issue to primary and secondary devices. TSO processes commands in the order that you issue them.

Table 42. Peer-to-peer remote copy TSO commands

TSO command	Can command be issued to a:	
	Primary device?	Secondary device?
CDELPAIR	Yes	Yes
CDELPATH	Yes	No
CESTPAIR	Yes	No

Table 42. Peer-to-peer remote copy TSO commands (continued)

TSO command	Can command be issued to a:	
	Primary device?	**Secondary device?**
CESTPATH	Yes	No
CGROUP	Yes	No
CQUERY	Yes	Yes
CRECOVER	No	Yes
CSUSPEND	Yes	Yes
PSETCHAR	Yes	No

Note: Do not use the NOCOPY option of the CESTPAIR command if you issue this command as part of automatically initiated PPRC subsystem activity, as this may introduce potential data integrity exposures.

You can put all PPRC TSO commands in a SYS1.PROCLIB member, in CLISTs, and include them within standard job JCL.

PPRC uses the Write-to-Operator (WTO) route code to route all messages to the console.

For additional information about TSO commands, refer to *z/OS TSO/E Command Reference*.

For additional information about the CESTPAIR NOCOPY option, refer to “CESTPAIR command syntax” on page 297.

CDELPAIR – deleting volume pairs

Use the CDELPAIR command to specify the primary and secondary volumes to remove from PPRC. Direct the command to the primary device.

For additional information about deleting pairs and example of the CLIST, refer to “Deleting a PPRC volume pair” on page 351.

CDELPAIR command syntax

The syntax of the CDELPAIR command is:

```
►►──CDELPAIR──DEVN──(──device_number──)──┬────────────────────┬──────────────►
                                         └─SUBCHSET(subchset)─┘

                                     (1)
►─PRIM──(──ssid serialno cca|lun lss──────)──────────────────────────────────►

                                    (2)
►─SEC──(──ssid serialno cca|lun lss──────)──┬──────────────────────────┬─────►◄
                                            │             ┌─NO──┐      │
                                            └─OPENDVCS(───┴─YES─┴──)───┘
```

Notes:

1 *lss* is required if the storage control supports logical subsystems (like the ESS)

2 *lss* is required if the storage control supports logical subsystems (like the ESS)

Required parameters

DEVN
Specifies the device number of the primary volume for the volume pair that PPRC is to delete. The device number is the 4-digit hexadecimal address of the device to which the I/O operation is directed.

If OPENDVCS(YES) is specified, DEVN identifies a System z device to act as a CKD access volume. This device must be located in the same subsystem cluster as the primary device.

PRIM Specifies the primary site storage subsystem ID (*ssid*), and serial number (*serialno*), the primary volume channel connection address (*cca*) or logical unit number (*lun*), and the logical storage subsystem (*lss*).

Where:

ssid Specifies a 4-character hexadecimal subsystem identifier (SSID). *ssid* must be specified in X'' notation. When OPENDVCS(YES) is specified, *ssid* can be the actual ssid, or X'FFFF' can be specified and the control unit will calculate the correct FB ssid.

serialno
Specifies the storage control serial number, which can include up to 12 digits, depending on the type of storage control.

cca Specifies a 2-digit channel connection address that is required when OPENDVCS(NO) is specified or allowed to default. *cca* must be specified in X'' notation.

lun Specifies the 2-digit logical unit number when OPENDVCS(YES) is specified. *lun* must be specified in X'' notation.

lss Specifies a 2-digit hexadecimal value that specifies the logical subsystem for the device (ESS only). *lss* must be specified in X'' notation.

Note: LSS number is required if the storage control supports logical subsystems (like the ESS) and not allowed if the storage control does not support logical subsystems.

SEC Specifies the recovery site storage subsystem ID and serial number, and the secondary volume channel connection address. The descriptions for the values are the same as for the PRIM parameter.

Optional parameters

OPENDVCS
Specifies whether Open System (fixed block) devices or CKD devices are addressed in the required parameters of this command. The values are:

NO The required parameters address CKD devices. This is the default.

YES Specifies that the required parameters in this command support open device management through a CKD access device.

SUBCHSET
Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as

defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is supported by the host system's processor and configured for the device.

If you omit the parameter, the command is issued to the device that is logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

CDELPAIR command examples

The following examples show the CDELPAIR command.

```
For CKD volumes:
  CDELPAIR DEVN(X'0C04') PRIM(X'6060' 62019 X'04')
    SEC(X'6061' 68006 X'09')

For CKD volumes in logical subsystems, such as ESS:

  CDELPAIR DEVN(X'0D20') PRIM(X'6114' 66143 X'08' X'06')
    SEC(X'6125' 68247 X'0B' X'03')

For open volumes:
  CDELPAIR DEVN(X'0E20') PRIM(X'FFFF' 68423 X'8C' X'16')
    SEC(X'FFFF' 68213 X'5D' X'17') OPENDVCS(YES)
```

CDELPATH – deleting paths

Use the CDELPATH command to delete all established paths between a primary site (source) logical subsystem and a recovery site (target) logical subsystem. Only active paths between the specified logical subsystems are affected; all other paths to other subsystems are unaffected.

Note:

1. Before issuing a CDELPATH command, issue a CDELPAIR command to all active PPRC volume pairs. The CDELPATH command might cause the issuance of an ANTP0121I message if you do not follow this sequence.
2. You can use the CESTPATH command to remove only selected paths. Each time you issue a CESTPATH command, the paths specified on it replace the paths that are established by the last CESTPATH command that was issued. This feature should be used with caution.

For additional information about deleting paths and a CLIST example, refer to “Deleting PPRC paths” on page 352.

For additional information about removing paths, refer to the note under “CESTPATH – establishing paths” on page 304.

CDELPATH command syntax

The syntax of the CDELPATH command is:

```
►►──CDELPATH──DEVN──(──device_number──)──┬─────────────────────┬──────────►
                                         └─SUBCHSET(subchset)─┘
```

```
                                                                   (1)
►─PRIM──(─ssid serialno|wwnn lss─)─SEC─(─ssid serialno|wwnn lss───────)──────►◄
```

Notes:

1 *lss* is required if the storage control supports logical subsystems (like the ESS)

Required parameters

DEVN
Specifies the device number of any disk volume that is attached to the primary site storage control without directing any operation to the volume. The device number is the 4-digit hexadecimal address of any disk volume behind the primary site storage control.

When a path between open device LSSs is being removed, DEVN identifies a System z device to act as a CKD access volume. This device must be located in the same subsystem cluster as the primary device.

PRIM For ESCON paths this parameter specifies the primary site storage subsystem ID (*SSID*), serial number (*serialno*), and logical subsystem (*LSS*).

Note: If the link addresses between subsystems use fibre channel port (FCP) numbers, WWNN is specified instead of storage control serial number.

Where:

ssid Specifies a 4-character hexadecimal subsystem identifier (SSID). *ssid* must be specified in X'' notation.

serialno
Specifies the storage control serial number, which can include up to 12 digits, depending on the type of storage control.

Note: If the link addresses between subsystems use fibre channel port (FCP) numbers, specify *wwnn* instead of *serialno*.

wwnn Specifies the World Wide Node Name in exactly 16-hexadecimal digits. The World Wide Node Name is used to access the primary storage control referenced by the *ssid* using FCP. When the *wwnn* is specified, it also indicates that the LINK addresses contain Fibre Channel adapter numbers. If you specify the *wwnn* for the primary logical subsystem, you must also specify the *wwnn* for the secondary logical subsystem.

NOTE: *wwnn* cannot be used with *serialno*.

lss Specifies a 2-digit hexadecimal value for the logical subsystem of the device (ESS only).

Note: LSS number is required if the storage control supports logical subsystems (like the ESS) and not allowed if the storage control does not support logical subsystems.

SEC Specifies the recovery site storage subsystem ID and serial number. The descriptions for the values are the same as for the PRIM parameter. **NOTE:** The PRIM and SEC parameters must both specify either a *serialno* or a *wwnn*.

Optional parameters

SUBCHSET

Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is supported by the host system's processor and configured for the device.

If you omit the parameter, the command is issued to the device that is logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

CDELPATH command examples

The following are examples of the CDELPATH command:

```
 CDELPATH DEVN(X'0C04') PRIM(X'6060' 62019)
   SEC(X'6061' 68006)

 CDELPATH DEVN(X'0D20') PRIM(X'6114' 66143 X'06')
   SEC(X'6125' 68247 X'07')

CDELPATH DEVN(X'0D20') PRIM(X'6114' 5005076300C11220 X'06')
  SEC(X'6125' 5005076300C33440 X'07')
```

CESTPAIR – establishing volume pairs

Use the CESTPAIR command to specify PPRC primary and secondary volumes. In order for this command to complete successfully, a PPRC link must be active between the primary logical subsystem and the secondary logical subsystem.

When establishing a pair of System z devices, the primary and secondary volumes must have the same number of tracks on each cylinder and the same number of bytes on each track. The secondary device must have the same number or a greater number of cylinders as compared to the primary device. For open devices, the secondary volume must have capacity equal to or greater than the primary volume.

Note:

1. Peer-to-peer remote copy allows you to copy data from a smaller primary device to a larger secondary device . (For System z volumes, the track size and number of tracks per cylinder must be the same.) The additional space on the secondary volume may not be immediately usable. End the PPRC volume pair and run ICKDSF Release 17 or above against the secondary volume to rebuild the VTOC and make the additional space usable.
2. Because multivolume data sets span multiple volumes, they require special handling. To ensure that the data set is recoverable, copy all volumes of multivolume and striped data sets.
3. For more information and examples, see "Adding a PPRC volume pair" on page 329.

CESTPAIR command syntax

The syntax of the CESTPAIR command is:

```
►►──CESTPAIR──DEVN──(──device_number──)──┬─────────────────────┬──────────►
                                         └─SUBCHSET(subchset)─┘

                                      (1)
►─PRIM──(──ssid serialno cca|lun lss─────)─────────────────────────────────►

►─SEC──(──ssid serialno cca|lun lss──)──┬──────────────────────────┬───────►
                                        │              ┌─NO──┐     │
                                        └─OPENDVCS(──┴─YES─┴──)─┘

►──┬─────────────────────────┬──┬───────────────────────────────┬──────────►
   │           ┌─NO──┐       │  │             ┌─SYNC─┐          │
   └─CASCADE(──┴─YES─┴──)──┘  └─OPTION──(──┴─XD───┴───────)─┘

►──┬────────────────────────────────┬──┬────────────────────────────────┬──►
   │           ┌─COPY────┐          │  │                  ┌─YES─┐       │
   └─MODE──(──┼─NOCOPY──┼───)──┘  └─AUTORESYNC──(──┴─NO──┴────)─┘
               ├─INCRES──┤
               └─RESYNC──┘

►──┬──────────────────────────────────┬──┬──────────────────────────┬──────►
   │                ┌─START──┐        │  │           ┌─NO──┐        │
   └─INCRESYNC──(──┼─STOP───┼───)──┘  └─FORCE──(──┴─YES─┴────)─┘
                    └─NOCOPY─┘

►──┬─PACE──(──15──)──────┬──┬───────────────────────┬──────────────────────►
   └─PACE──(──1–255──)──┘  │          ┌─NO──┐      │
                            └─CRIT──(──┴─YES─┴───)─┘

►──┬────────────────────────────┬──┬─ACTION(──┬─FAILOVER───┬──)─┬──────────►
   │             ┌─NO──┐        │             ├─FAILBACK───┤
   └─ONLINSEC──(──┴─YES─┴───)─┘             └─MTFAILOVER─┘

►──┬─────────────────────────┬──┬──────────────────────────┬───────────────►
   │           ┌─NO──┐       │  │            ┌─NO──┐       │
   └─MSGREQ──(──┴─YES─┴───)─┘  └─SETGTOK──(──┴─YES─┴───)─┘

►──┬────────────────────────────┬──────────────────────────────────────────►◄
   │             ┌─NO──┐        │
   └─RESETRSV──(──┴─YES─┴───)─┘
```

Notes:

1 *lss* is required if the storage control supports logical subsystems (like the ESS)

Required parameters

DEVN
Specifies the device number of the primary volume for the volume pair to be established. The device number is the 4-digit hexadecimal address of the device to which the I/O operation is directed.

If OPENDVCS(YES) is specified, DEVN identifies a System z device to act as a CKD access volume. This device must be located in the same subsystem cluster as the primary device.

PRIM Specifies the primary site storage subsystem ID (*ssid*), and serial number (*serialno*), the primary volume channel connection address (*cca*), and the logical storage subsystem (*lss*).

Where:

ssid Specifies a 4-character hexadecimal subsystem identifier (SSID). *ssid* must be specified in X'' notation. When OPENDVCS(YES) is specified, *ssid* can be the actual ssid, or X'FFFF' can be specified and the control unit will calculate the correct FB ssid.

serialno
Specifies the storage control serial number, which can include up to 12 digits, depending on the type of storage control.

cca Specifies a 2-digit channel connection address in X'' notation.

lun Specifies the 2-digit logical unit number when OPENDVCS(YES) is specified. *lun* must be specified in X'' notation.

lss Specifies a 2-digit hexadecimal value that specifies the logical subsystem for the device (ESS only).

Note: LSS number is required if the storage control supports logical subsystems (like the ESS) and not allowed if the storage control does not support logical subsystems.

SEC Specifies the recovery site storage subsystem ID and serial number, and the secondary volume channel connection address. The descriptions for the values are the same as for the PRIM parameter.

Optional parameters

OPENDVCS
Specifies whether Open System (fixed block) devices or CKD devices are addressed by this command. The values are:

NO The required parameters address CKD devices. The default is NO.

YES Specifies that the required parameters in this command target fixed block devices using a CKD access device.

CASCADE
Specifies whether primary volume in this pair is also eligible to be the secondary volume in another PPRC pair. The values are:

NO The primary volume is not eligible to be secondary in another PPRC pair. The default is NO.

YES The primary volume is eligible to be secondary in another PPRC pair.

Note: CASCADE(YES) is not valid with the ACTION parameter.

Note: CASCADE(YES) is not valid with CRIT(YES) or the PACE parameter.

FORCE
Specifies whether validation of the volumes involved in the establish request occurs or is bypassed. This parameter is only valid when specified with MODE(INCRES), ACTION (FAILOVER), or ACTION (FAILBACK). **No** is the default.

YES

- Specifying FORCE(YES) with MODE(INCRES) indicates that normal validation that the volumes in the request have an intermediate volume in common is bypassed.
- Specifying FORCE(YES) with ACTION(FAILOVER) or ACTION(FAILBACK) indicates that validation that the specified secondary is the existing suspended device in the existing relationship for the specified primary in FAILOVER mode is bypassed.

NO

- Specifying FORCE(NO) with MODE(INCRES) indicates the following validation:
  - The current failed secondary device for the primary specified, and the current failed primary device for the secondary specified are the same device.
- Specifying FORCE(NO) with ACTION(FAILOVER) or ACTION(FAILBACK) indicates the following validation:
  - The devices specified in the establish request are currently part of a suspended relationship (in failover state for ACTION(FAILBACK)).

AUTORESYNC
Specifies whether the auto resync capability is to be enabled or disabled for the Global Copy pair. The auto resync capability attempts to automatically resynchronize the pair if it has been suspended due to something other than a command. The values are:

YES
The auto resync capability should be enabled for the Global Copy pair. This is the default.

NO The auto resync capability should be disabled for the Global Copy pair.

You may want to use NO to disable the auto resync capability when the secondary of the Global Copy pair is or will be the source of a Space Efficient FlashCopy relationship. An outage that results in many updates that must be copied when the pair is resynchronized could cause the space efficient repository to reach warning levels or even be filled.

INCRESYNC
Specifies whether the incremental resync change recording mechanism for Metro/Global Mirror should be enabled. There is no default. INCRESYNC is mutually exclusive with ACTION(FAILOVER).

START
Specifies the change recording mechanism should be started.

STOP Specifies the change recording mechanism should be stopped

NOCOPY

Used in recovery scenarios, where the incremental resync change recording mechanism needs to be started. INCRESYNC(NOCOPY) indicates that the change recording bitmaps do not need to be initialized with all bits on. Thus, when MODE(INCRES) is used to establish the primary device to a different secondary, a full copy is not done. The reason is that in the recovery environment where this is used, the intended devices are already in sync, except for the changes made after the incremental resync change recording mechanism is started.

Note: If the INCRESYNC(NOCOPY) parameter is specified in scenarios other than where it is intended, this could cause you to make the recovery environment vulnerable to data integrity exposures.

MODE

Specifies one of the following PPRC modes:

COPY Specifies that PPRC copy all tracks on the primary volume to the secondary volume. COPY is the default mode.

Note: MODE(COPY) is not valid with the ACTION parameter.

NOCOPY

Specifies that only those tracks on the primary volume that are updated after this command is issued are to be copied to the secondary volume. Specify this mode only if the volumes are *exact* copies of each other.

The NOCOPY option assumes that the specified volumes are fully synchronized at the time the copy is started. Ensure that no I/O operations occur to either volume prior to starting the copy operation. This is necessary in order to protect the data integrity of the two volumes.

The following notes apply only when the volumes of a PPRC volume pair are not of the same device capacity (volumes do not have the same number of cylinders):

For PPRC CKD Volume Pairs of Different Device Capacity:

1. If you have established or reestablished this PPRC volume pair without specifying COPY, run ICKDSF REFORMAT REFVTOC on the primary volume after you establish the pair. Doing so allows the VTOC and indexed VTOC information to accurately represent the primary volume.
2. If you switch the primary and secondary of this pair at any time, run ICKDSF REFORMAT REFVTOC on the new primary volume. Doing so allows the VTOC and indexed VTOC information to accurately represent the new primary volume.
3. If you remove the secondary volume from the volume pair for use as a simplex volume, you should run ICKDSF REFORMAT REFVTOC on the volume. Doing so allows the VTOC and Indexed VTOC information to accurately represent the secondary volume.

INCRES

Causes an incremental resynchronization using the change recording bitmaps to limit the tracks to be copied. MODE(INCRES) is mutually exclusive with the ACTION keyword and with the INCRESYNC keyword.

RESYNC
Specifies that PPRC reestablish a suspended copy. Before you issue this option, be certain that either the primary and secondary volumes are both in a suspended state, or the secondary is in a simplex state and the primary is suspended. PPRC only copies from the primary volume to the secondary volume the cylinders that were updated during the period of suspension.

OPTION
Specifies the option used for transmitting data to PPRC secondary volumes. These options can be used to convert from one PPRC state to another. For more information, see "Converting from one PPRC volume state to another" on page 356. The values are:

SYNC Specifies that data is transmitted synchronously to PPRC secondary volumes. SYNC is the default value. You can specify this option for all storage controls that support PPRC operations. Specifying OPTION(SYNC) will cause a PPRC extended distance pair in PENDING XD or SUSP(*n*).XD state to change to a synchronous operation.

XD Specifies that data is transmitted from primary volumes to secondary volumes in PPRC extended distance mode. The IBM ESS must have the PPRC extended distance feature installed and active for this option to be accepted. The CRIT(YES) and MODE(NOCOPY) parameters are mutually exclusive with the OPTION(XD) parameter.

For information about PPRC extended distance, see "Managing PPRC extended distance mode" on page 353.

Note: OPTION(XD) is not valid with CRIT(YES). OPTION(XD) is not valid with MODE(NOCOPY) if you also specify ACTION(FAILOVER) or ACTION(FAILBACK).

ONLINSEC
Specifies whether the secondary volume attached to an ESS is online to a host system. The values are:

NO Specifies that, if the secondary volume that is attached to an ESS subsystem appears to be online to any host system, the CESTPAIR command will fail with an error. The default is NO.

YES Specifies that the CESTPAIR command be processed even if the ESS secondary volume appears to be online to a host system.

PACE Specifies the minimum number of tracks that PPRC is to copy at one time before allowing another host interrupt. The allowable range is from 1 to 255 tracks. If you omit this parameter, PPRC uses the storage control default value. PPRC ignores the PACE parameter if you also specify the NOCOPY parameter. PPRC ignores the PACE parameter with storage subsystems that support logical storage subsystems (LSS).

PACE is ignored when OPENDVCS(YES) is specified.

Note: PACE is not valid with the ACTION parameter.

Table 43 shows the result of different PACE parameter values.

Table 43. PACE parameter values

Value	Effect on the CESTPAIR copy operation
1	Copy a maximum of three tracks at a time
2 to 255	Copy a maximum of 15 tracks at a time

CRIT Specifies whether PPRC allows subsequent write requests to the primary volume. The values are:

NO Specifies that, following an I/O completion error, PPRC allows subsequent write requests to the PPRC primary volume. The primary storage control will perform change recording. The default is NO.

YES This should only be specified at the request of an IBM representative. Specifies that if an I/O error occurs, PPRC either allows or does not allow subsequent writes, depending on how the storage subsystem is configured. The PPRC pair then remains in a suspended state until you correct the problem and issue either a CESTPAIR RESYNC or CDELPAIR command. See "Considering the PPRC solution" on page 267 and "Examining disaster recovery from a system viewpoint" on page 367 for related information.

Note: CRIT(YES) is not valid with OPTION(XD), CASCADE(YES), the ACTION parameter, or with OPENDVCS(YES).

ACTION

The ACTION option is valid only for existing PPRC pairs and is used to temporarily failover processing to the recovery site when an outage occurs at the primary site, and to restore processing at the primary site when it is capable of resuming operation. The values are:

FAILOVER

Specifies that the direction of the PPRC pair is to be reversed and change recording is to begin. Upon completion of the command, the device that was previously the secondary becomes the suspended primary device in the PPRC pair. Change recording begins on the suspended primary, to be used in a subsequent failover operation. ACTION(FAILOVER) is not valid with MSGREQ(YES).

FAILBACK

Specifies that the connection between the primary and secondary devices has been repaired and the original primary/secondary relationship is to be restored. Changed tracks recorded since failover was issued are copied to the specified secondary device.

If the secondary is in a soft fence state, the soft fence state is cleared.

MTFAILOVER

Specifies that a multi-target configuration should be created from a cascaded configuration. For example, in a cascaded configuration, H1->H2->H3, where H1 is the primary, issuing a CESTPAIR command with ACTION(MTFAILOVER) to the middle device, H2, causes a multi-target configuration (H2->H1 and H2-H3) in place of the previous cascaded failover state.

For more information, refer to "Converting to Multi-Target Mirror" on page 275.

Note: ACTION is not valid with MODE(COPY), CASCADE(YES), CRIT(YES) or with PACE.

MSGREQ
The MSGREQ option is valid only with the COPY option. The values are:

NO Specifies that the command is complete as soon as you initiate the request to the storage control. PPRC does not wait for the copy operation to complete before issuing completion messages. The default is NO.

YES Specifies that PPRC wait until the initial full-volume copy operation is complete before issuing completion message ANTP0001I. This option is not valid when OPENDVCS(YES) or ACTION(FAILOVER) is specified.

RESETRSV
Indicates whether an existing reserve on the specified fixed block secondary device should be reset (removed) by the control unit during the establish-pair processing. The values are:

NO Any reserve that exists on the specified secondary device should not be reset. This is the default.

YES If there is a reserve on the specified secondary device, that reserve should be reset.

SETGTOK
Indicates whether the secondary volume of the specified pair is a space-efficient volume. The values are:

NO Indicates that secondary cannot be a space efficient volume. The default value is NO.

YES Indicates that the secondary volume can be a space efficient volume. If YES is specified and the secondary volume is not a space efficient volume, the command is processed as if the keyword was not specified.

SUBCHSET
Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is supported by the host system's processor and configured for the device.

If you omit the parameter, the command is issued to the device that is logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

CESTPAIR command examples

The following examples show the use of the CESTPAIR command:

```
For CKD volumes:
    CESTPAIR DEVN(X'C602') PRIM(X'C600' 24886 X'02' X'06') SEC(X'7300' 46807 X'09' X'03')
             OPTION(XD) MODE(COPY) CRIT(NO)

    CESTPAIR DEVN(X'F50') PRIM(X'BAC0' 01711 X'14' X'06') SEC(X'7E00' 22243
X'18' X'06')
             ACTION(FAILOVER) ONLINSEC(YES)
             ANTP0001I CESTPAIR COMMAND COMPLETED FOR DEVICE 0F50. COMPLETION CODE: 00

    CESTPAIR DEVN(X'F50') PRIM(X'BAC0' 01711 X'14' X'06') SEC(X'7E00' 22243
X'18' X'06')
             ACTION(FAILBACK) ONLINSEC(YES)
             ANTP0001I CESTPAIR COMMAND COMPLETED FOR DEVICE 0F50. COMPLETION CODE: 00
```

```
For open volumes:
    CESTPAIR DEVN(X'0240') PRIM(X'FFFF' 42685 X'0A' X'10') SEC(X'FFFF' 22940 X'4D' X'18')
             OPENDVCS(YES) MODE(COPY)

    CESTPAIR DEVN(X'F50') PRIM(X'FF21' 01711 X'06' X'21') SEC(X'FF22' 22243 X'04' X'16')
             OPENDVCS(YES) ACTION(FAILOVER)
             ANTP0001I CESTPAIR COMMAND COMPLETED FOR DEVICE 0F50. COMPLETION CODE: 00

    CESTPAIR DEVN(X'F50') PRIM(X'FF21' 01711 X'06' X'21') SEC(X'FF22' 22243 X'04' X'16')
             OPENDVCS(YES) ACTION(FAILBACK)
             ANTP0001I CESTPAIR COMMAND COMPLETED FOR DEVICE 0F50. COMPLETION C0DE: 00
```

CESTPATH – establishing paths

Use the CESTPATH command to establish ESCON or FCP (fibre channel protocol) paths between a primary site (source) logical subsystem and a recovery site (target) logical subsystem, or to establish control paths between a master storage control and storage control subordinates in a Global Mirror session.

Each CESTPATH command can establish up to 8 paths (if the primary site logical subsystem supports eight paths) from one primary site logical subsystem to a single recovery site logical subsystem. You can link up to 16 recovery site logical subsystems to one primary site logical subsystem. These links require that you issue a separate CESTPATH command for each logical subsystem.

When you designate the use of Fibre Channel Protocol (FCP) the logical subsystem serial number (*serialno*) is replaced by the logical subsystem World Wide Node Name (*WWNN*). Designating FCP affects the following parameters:

- PRIM
- SEC
- LINK

Attention: The CESTPATH command is a “replace” command. The paths on the command are treated as a path set. Any subsequent CESTPATH command removes paths previously established and replaces them with the new set of paths. You can use the CESTPATH command to selectively add and remove paths. However, you cannot use this command to replace paths if you are trying to switch from ESCON to FCP or vice versa.

Note: Use caution when using this command to ensure that you do not remove paths that you want to keep.

If you are trying to switch all your paths in a LSS to LSS association from ESCON to FCP or vice versa, you can use the following procedure:

1. Issue a CGROUP FREEZE command to the primary LSS. This will remove all of the ESCON paths and suspend all the PPRC pairs in the LSS to LSS association.
2. Issue a CGROUP RUN command to allow Host application I/O to the primary volumes to resume.
3. Issue the CESTPATH command using the applicable required parameters to re-establish the PPRC paths.
4. Issue the CESTPAIR command with the MODE(RESYNC) option to re-establish the PPRC pairs.

The order in which links are added or deleted is determined by the subsystem.

CESTPATH command syntax

The syntax of the CESTPATH command is:

```
►►──CESTPATH──DEVN──(──device_number──)──┬──────────────────────┬──────────────►
                                         └─SUBCHSET(subchset)──┘

                                                                   (1)
►─PRIM──(──ssid serialno|wwnn lss──)──SEC──(──ssid serialno|wwnn lss──)──────────►

            ┌──────────┐
►─LINK──(───▼─linkaddr─┴───)──┬─────────────────────────────┬──────────────────►
                              │               ┌─No──┐        │
                              └─CGROUP──(─────┴─Yes─┴────)──┘

►──┬──────────────────────────────┬──────────────────────────────────────────►◄
   │               ┌─No──┐        │
   └─RESETHP──(────┴─Yes─┴────)──┘
```

Notes:

1 *lss* LSS number is required if the storage control supports logical subsystems (like the ESS)

Required parameters

DEVN
: Specifies the device number of any simplex disk volume that is attached to the primary site storage control without directing any operation to the volume. The device number is the 4-digit hexadecimal address of any disk volume behind the primary site storage control.

  When defining paths between open device LSSs, DEVN identifies a System z device to act as a CKD access volume. This device must be located in the same subsystem cluster as the primary device.

PRIM For ESCON paths this parameter specifies the primary site subsystem ID (*SSID*), serial number (*serialno*), and logical subsystem (*LSS*). For FCP paths, this parameter specifies the primary site subsystem ID (*SSID*), the World Wide Node Name (*wwnn*), and logical subsystem (*LSS*).

Where:

ssid
: Specifies a 4-character hexadecimal subsystem identifier (SSID). *ssid* must be specified in X'' notation.

serialno
: Specifies the storage subsystem serial number, which can include up to 12 digits, depending on the type of logical subsystem. Using a serial number indicates the LINK addresses are for ESCON channel interfaces. If the serial number is supplied for the primary logical subsystem, the serial number must also be supplied for the secondary logical subsystem.

  Note: If the link addresses between subsystems use fibre channel port (FCP) numbers, WWNN is specified instead of storage control serial number.

wwnn
: Specifies the World Wide Node Name and it is exactly 16-hexadecimal digits in length. The World Wide Node Name is used to access the primary logical subsystem referenced by the *ssid* using FCP. When the *wwnn* is specified it also indicates that the LINK addresses contain Fibre Channel adapter numbers. If you specify the *wwnn* for the primary logical subsystem, you must also specify the *wwnn* for the secondary logical subsystem. **NOTE:** *wwnn* cannot be used with *serialno*.

lss
: Specifies a 2-digit hexadecimal value that specifies the logical subsystem for the device (ESS only).

Note: LSS number is required if the storage control supports logical subsystems (like the ESS) and not allowed if the storage control does not support logical subsystems.

SEC
: Specifies the recovery site storage subsystem ID and serial number. The descriptions for the values are the same as for the PRIM parameter.

  Note: The PRIM and SEC parameters must both specify either a *serialno* or a *wwnn*.

LINK
: Specifies the addressing paths from the primary site logical subsystem to the recovery site logical subsystem.

  Note: Duplicate link addresses are not allowed.

  You can specify up to eight path addresses (if the primary site logical subsystem supports eight paths), separated with a blank, for each LINK parameter.

  Each path address is an 8-digit hexadecimal address. If *serialno* is specified on the PRIM and SEC parameters, the format of each path address has the hex digit form of *aaaabbcc* where:

  aaaa
  : Specifies the System Adapter ID (SAID) interface identifier on the primary system that uniquely identifies the physical location of the associated storage subsystem interface. The first two values are zeros (00*aa*), and the last two values are hexadecimal SAID byte values that correspond to ESCON channel interfaces, as follows:

```
bit 0 = storage cluster (0 or 1)
bit 1 = 0
bit 2 = I/O board (0 or 1)
bit 3 = 0
bits 4-5 = slot (00, 01, 10, or 11)
bit 6 = 0
bit 7 = port number (0 or 1 for each slot)
```

bb Specifies the recovery site logical subsystem ESCON link destination address. If the connection is routed through an ESCON Director, the bb value is the ESCON Director destination port. This value will be 00 if the link is between two open device logical subsystems. For direct fiber-optic cable connections between storage controls or static connections through an ESCON Director, the link address value is zero.

cc For storage controls supporting logical subsystems, this value is the logical subsystem number. For storage controls not supporting logical subsystems or for fixed block logical subsystems, this value is 00.

If *wwnn* is specified on the PRIM and SEC parameters, the format of each Fibre Channel path address has the hex digit form of *aaaabbbb* where:

aaaa Specifies the Fibre Channel adapter used for the path in the primary subsystem.

bbbb Specifies the Fibre Channel adapter used for the path in the secondary subsystem.

For additional information about the LINK parameter, see "Establishing PPRC paths" on page 325.

Optional parameters

CGROUP

Specifies whether a PPRC volume pair in a consistency group remains in a long-busy condition for a specified period following an error that is related to the associated volume pair. When an error occurs and one of the volumes in the consistency group becomes suspended, volumes in that consistency group become suspended the next time they are written to.

You must specify one of the following values with the CGROUP parameter:

NO Specifies that if an error occurs to a PPRC volume pair in a consistency group, the long-busy condition is not enabled for the associated volume pair. NO is the default.

YES Specifies that if an error occurs to a PPRC volume pair in a consistency group, the long-busy condition is enabled for the associated volume pair. Automation products can detect PPRC volume pair outages on these devices and take appropriate action.

RESETHP

At this time, only the ESS subsystem supports the RESETHP option. This parameter is ignored when you have specified *wwnn* on the PRIM and SEC parameters.

You must specify one of the following values with the RESETHP parameter:

NO Specifies that PPRC *not* reset existing host system paths to the primary subsystem before establishing links to the secondary subsystem. The default is NO.

YES Specifies that PPRC reset existing host system paths to the primary subsystem before establishing links to the secondary subsystem.

SUBCHSET
Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is supported by the host system's processor and configured for the device.

If you omit the parameter, the command is issued to the device that is logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

CESTPATH command examples

The following are examples of the CESTPATH command:

```
   CESTPATH DEVN(X'0142') PRIM(X'6060' 62019)
SEC(X'6061' 68006)
              LINK(X'0004A100')

   CESTPATH DEVN(X'D200') PRIM(X'D200' 32178 X'02')
 SEC(X'C700' 22831 X'07')
              LINK(X'0028E007') RESETHP(YES)

   CESTPATH DEVN(X'0D20') PRIM(X'6114' 66143 X'06')
 SEC(X'6125' 68247 X'07')
              LINK(X'0004A100') CGROUP(YES)

 CESTPATH DEVN(X'A347') PRIM(X'A300' 5005076300C56560 X'03')
SEC(X'D400' 5005076300C63584 X'04')
              LINK(X'00240028') CGROUP(YES)
```

CGROUP – controlling volume groups

Issue the CGROUP command to control operations for multiple PPRC volume pairs on a logical subsystem or single storage control. This command lets you suspend or resume all operations for all PPRC volumes on a logical subsystem or single storage control. You must issue a separate CGROUP command to suspend or resume operations on each logical subsystem or storage control.

CGROUP command syntax

The syntax of the CGROUP command is:

```
►►──CGROUP──DEVN──(──device_number──)──┬─────────────────────┬──────────────►
                                       └─SUBCHSET(subchset)─┘

                                                       (1)
►─PRIM──(──ssid serialno lss──)──SEC──(──ssid serialno lss──)──────┬────────┬──►◄
                                                                   ├─FREEZE─┤
                                                                   └─RUN────┘
```

Notes:

1 *lss* LSS number is required if the storage control supports logical subsystems (like the ESS)

Required parameters

DEVN
: Specifies the device number of any disk volume that is attached to the primary site storage control without directing any operation to the volume. The device number is the 4-digit hexadecimal address of any disk volume behind the primary site storage control.

  When issuing this command for an open device LSS, DEVN specifies a System z device located in the same subsystem cluster as the primary device.

PRIM Specifies the primary site storage subsystem ID (*SSID*), serial number (*serialno*), and logical storage subsystem (*LSS*).

Where:

ssid Specifies a 4-character hexadecimal subsystem identifier (SSID). *ssid* must be specified in X'' notation.

serialno
: Specifies the storage subsystem serial number, which can include up to 12 digits, depending on the type of logical subsystem. Using a serial number indicates the LINK addresses are for ESCON channel interfaces. If the serial number is supplied for the primary logical subsystem, the serial number must also be supplied for the secondary logical subsystem.

lss Specifies a 2-digit hexadecimal value that identifies the logical subsystem for the device.

Note: LSS number is required if the storage control supports logical subsystems (like the ESS) and not allowed if the storage control does not support logical subsystems.

SEC Specifies the recovery site storage subsystem ID and serial number. The description for the values is the same as for the PRIM parameter.

Optional parameters

FREEZE
: Specifies that PPRC stop all updates to secondary subsystems for pairs that have primary and secondary volumes on the primary and secondary logical subsystems identified in the CGROUP command. The system suspends these pairs and all paths between the identified primary and secondary subsystems removes.

  Use the following guidelines; all host application I/O to the primary is held off with a long busy condition for two minutes, or until a CGROUP command is issued with the RUN option.

  - Establish the path with CGROUP(YES) specified in the CESTPATH command.
  - Establish the paths with CGROUP(YES) specified in the copy services GUI or CLI when the consistency group runs.

  If you do not specify CGROUP(YES) in the CESTPATH command, the host I/O is not held off with a long busy condition. To resume operations you must re-establish the paths between the primary and secondary subsystems, and then re-establish the pairs using the MODE(RESYNC) parameter.

To use FREEZE for the consistency of all volumes in a group after an error, use the following process:

- Specify CESTPATH command with CGROUP(YES) when establishing paths carrying data for the group.
- Use CGROUP with FREEZE on all affected primary to secondary LSS pairs.
- Use CGROUP with FREEZE on all affected primary to secondary LSS pairs before the control unit timer has elapsed.

Note:

1. PPRC does not allow any activity to the secondary volumes.
2. You cannot specify the FREEZE parameter with the RUN parameter.
3. Issue all the FREEZE commands before issuing the RUN commands.

RUN Specifies that host application I/O may resume on primary volumes for pairs which have primary and secondary volumes on the primary and secondary logical subsystems identified in the CGROUP command which have been suspended due to an earlier CGROUP FREEZE command. The pairs remain suspended. You cannot specify the RUN parameter with the FREEZE parameter.

SUBCHSET
Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is supported by the host system's processor and configured for the device.

If you omit the parameter, the command is issued to the device that is logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

CGROUP command examples

The following are examples of the CGROUP command:

```
CGROUP DEVN(X'0F40') PRIM(X'6060' 62019) SEC(X'6061' 68006) FREEZE

CGROUP DEVN(X'OF40') PRIM(X'6060' 62019) SEC(X'6061' 68006) RUN

CGROUP DEVN(X'4683') PRIM(X'4600' 62019 X'06') SEC(X'B600' 68006 X'06') FREEZE

CGROUP DEVN(X'4683') PRIM(X'4600' 62019 X'06') SEC(X'B600' 68006 X'06') RUN
```

CQUERY – querying status

Use the CQUERY command to query the status of one volume of a PPRC volume pair, or all paths that are associated with the logical subsystem or storage control for the device number that you specify. You can also display the linkage adapter information between two controllers. You can issue the CQUERY command to either a primary or secondary PPRC volume. A host system that is attached only to a primary volume cannot obtain the status of the secondary volume for that pair. In the same way, a host attached only to the secondary volume cannot obtain the status of the primary volume.

See "Querying PPRC volumes" on page 330 and "Querying PPRC paths" on page 345 for CQUERY command output examples for a volume, and for paths.

CQUERY command syntax

The syntax of the CQUERY command is:

```
►►──CQUERY──DEVN(device_number)──┬──────────────────────┬──────────────►
                                 └─SUBCHSET(subchset)─┘

►─┬──────────────────────────┬─────────────────────────────────────────►
  └─ODEVN(─serno lun lss─)─┘

►─┬───────────────────────────────────────────────────────────────┬────►
  │             ┌─BITMAP───┐                                       │
  ├─VOLUME──────┴─NOBITMAP─┴───────────────────────────────────────┤
  │      (1)                                                       │
  ├─PATHS──────────────────────────────────────────────────────────┤
  └─LINKINFO──┬─SDEVN(device_number)──┬──────────────────────┬─┬───┘
              │                       └─SSUBCHSET(ssubchset)─┘ │
              └─SWWNN(world-wide-node-name)────────────────────┘

►─┬────────────┬───────────────────────────────────────────────────────►◄
  │ ┌─FORMAT───┐│
  └─┴─UNFORMAT─┴┘
```

Notes:

1 BITMAP and NOBITMAP are ignored if you specify the PATHS parameter.

Required parameters

DEVN

Specifies the device number of the primary volume that PPRC is to query. The device number is the 4-digit hexadecimal address of the device to which the I/O operation is directed.

When issuing this command for an open device LSS, DEVN specifies a System z device located in the same subsystem cluster as the device being queried.

Optional parameters

ODEVN

Specifies address information for the open device to be queried. (If CKD device address information is specified with this parameter, CKD data will be returned.) The values are:

serno Specifies the storage subsystem serial number, which can include up to 12 digits depending on the type of logical subsystem.

lun Specifies the 2-digit logical unit number. *lun* must be specified in X'' notation.

lss Specifies a 2-digit hexadecimal value that identifies the logical subsystem for the device.

VOLUME

Directs PPRC to display the status of the volume that is specified with the DEVN parameter. The default value is VOLUME.

BITMAP

Specifies that PPRC report the percentage of tracks that it has copied for a pending or suspended PPRC primary volume. This option is valid when you query a PPRC primary volume and you either specify the VOLUME keyword, or you do not specify any optional parameters. BITMAP is ignored if you specify the PATHS keyword. If you specify the VOLUME keyword, the default value is BITMAP.

NOBITMAP

Specifies that PPRC not report the percentage of tracks that it has copied for a pending or suspended PPRC primary volume. This option is valid when you query a PPRC primary volume and you either specify the VOLUME keyword, or you do not specify any optional parameters.

You can use the NOBITMAP option to collect some data from a storage control even when the device is unable to respond to the CQUERY command.

Example: PPRC may not be able to collect bitmap information because the device is disabled.

PATHS

Directs PPRC to display all of the paths that are associated with the primary site storage control, and the status of each path. If you direct this command to the secondary device, PPRC cannot make path or storage control information available. To obtain this information, issue the command to a volume on that storage control that is either in simplex state or is a primary of a duplex pair.

FORMAT

Specifies that PPRC present the CQUERY command output in a block format. For examples refer to “Examples of CQUERY output” on page 336. The default value is FORMAT.

PPRC issues message ANTP0090I as part of a block report when you specify the CQUERY VOLUME with FORMAT option. PPRC issues message ANTP0095I as part of a block report when you specify the CQUERY PATH with FORMAT option.

UNFORMAT

Specifies that PPRC present the CQUERY command output as in the examples in “CQUERY unformatted output for a primary volume” on page 338 for a volume query and “CQUERY unformatted output with the PATHS option” on page 346 for a paths query.

PPRC issues message ANTP0091I as part of a block report when you specify the CQUERY VOLUME with UNFORMAT option.

PPRC issues message ANTP0096I as part of a block report when you specify the CQUERY PATHS with UNFORMAT option.

LINKINFO

Requests that PPRC display all of the potential connectivity of the Fibre

Channel ports in the storage facility image (SFI) specified by the DEVN parameter to each system adapter port in the SFI specified by the SWWNN or SDEVN parameter.

SDEVN
Specifies the device number of the secondary volume that PPRC is to query. The device number is the 4-digit hexadecimal address of the device to which the I/O operation is directed.

SWWNN
Specifies the world wide node name of the secondary volume that PPRC is to query. The world wide node name is the 8-byte hexadecimal identifier of the SFI to which the I/O operation is directed.

SUBCHSET
Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is supported by the host system's processor and configured for the device.

If you omit the parameter, the command is issued to the device that is logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

SSUBCHSET
Specifies the subchannel set for the secondary device specified with the SDEVN parameter. The subchannel set is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The values are determined by what is supported by the host system's processor and configured for the device.

If you omit the parameter, the command is issued to the device that is logically in subchannel set 0.If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

CQUERY command examples

The following are examples of the CQUERY command:

```
CQUERY DEVN(X'0C04') PATHS UNFORMAT

CQUERY DEVN(X'0C04') VOLUME

CQUERY DEVN(X'0C04') NOBITMAP

CQUERY DEVN(X'0C04') VOLUME BITMAP

CQUERY DEVN(X'0E0D') ODEVN(27422 X'8C' X'17')
```

CRECOVER – recovering data on the recovery system

Use the CRECOVER command to allow the recovery system to gain control of a disk volume on its logical subsystem or storage control. This command is issued from the recovery system. It signals the recovery site logical subsystem or storage control to force the secondary volume into simplex state to establish recovery system control. During this process, verify the volume serial number, and relabel the volume if you desire. You can vary the volume online after this command function is complete.

CRECOVER command syntax

The syntax of the CRECOVER command is:

```
►►──CRECOVER──DEVN──(──device_number──)──┬──────────────────────┬──────────────►
                                         └─SUBCHSET(subchset)─┘

►─PRIM──(──ssid serialno cca|lun lss──)────────────────────────────────────────►

                                      (1)
►─SEC──(──ssid serialno cca|lun lss──)───┬──────────────────────────┬──────────►
                                         │             ┌─NO──┐      │
                                         └─OPENDVCS(──┴─YES─┴──)─┘

►──┬─────────────────────────────────────────┬─────────────────────────────────►◄
   └─ID──(──old volser──┬─────────────┬──)─┘
                        └─new volser─┘
```

Notes:

1 *lss* LSS number is required if the storage control supports logical subsystems (like the ESS)

Required parameters

DEVN
: Specifies the secondary volume that is attached to the recovery site host. The device number is the 4-digit hexadecimal address of the device to which the I/O operation is directed.

  If OPENDVCS(YES) is specified, DEVN identifies a System z device to act as a CKD access volume. This CKD device must be located in the same subsystem cluster as the secondary device.

PRIM
: Specifies the primary site storage subsystem ID (*ssid*), and serial number (*serialno*), the primary volume channel connection address (*cca*), and the logical storage subsystem (*lss*).

  Where:

  ssid
  : Specifies a 4-character hexadecimal subsystem identifier (SSID). *ssid* must be specified in X'' notation. When OPENDVCS(YES) is specified, *ssid* can be the actual ssid, or X'FFFF' can be specified and the control unit will calculate the correct FB ssid.

  serialno
  : Specifies the storage control serial number, which can include up to 12 digits, depending on the type of storage control.

  cca
  : Specifies a 2-digit channel connection address. *cca* must be specified in X'' notation.

  lun
  : Specifies the 2-digit logical unit number when OPENDVCS(YES) is specified. *lun* must be specified in X'' notation.

lss Specifies a 2-digit hexadecimal value that specifies the logical subsystem for the device (ESS only).

Note: LSS number is required if the storage control supports logical subsystems (like the ESS) and not allowed if the storage control does not support logical subsystems.

SEC Specifies the recovery site storage subsystem ID and serial number, and the secondary volume channel connection address. The descriptions for the values are the same as for the PRIM parameter.

Optional parameters

OPENDVCS
Specifies whether Open System (fixed block) devices or CKD devices are addressed in the required parameters of this command. The values are:

NO The required parameters address CKD devices. The default is NO.

YES Specifies that the required parameters in this command target fixed block devices using a CKD access device.

ID Specifies that volume serial verification is performed when you use this parameter. You can specify a 1–6 character volume serial number for this parameter. If you specify the ID parameter, the *old volser* is required and the *new volser* is optional. If you do not specify the ID parameter, volume serial verification is *not* performed.

ID is not valid when OPENDVCS(YES) is specified.

Note: The secondary volume is reset to SIMPLEX state regardless of the following conditions:

- The ID parameter is not specified
- The volume serial numbers specified do not match
- A new volser is specified but the volume relabeling operation fails

The values are:

old volser
Specifies the 1–6 character volume serial number that will be compared to the PPRC secondary volume serial number. If the *old volser* does not match the secondary volume serial number, an error message is issued.

new volser
Specifies the 1–6 character volume serial number that will replace the PPRC secondary volume serial number if the *old volser* matches the secondary volume serial. If the *old volser* does not match the secondary volume serial number, this parameter is ignored.

SUBCHSET
Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is supported by the host system's processor and configured for the device.

If you omit the parameter, the command is issued to the device that is logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

CRECOVER command example

The following is an example of the CRECOVER command :

```
For CKD device:
     CRECOVER DEVN(X'0C04') PRIM(X'6060' 62019 X'04' ) SEC(X'6061' 68006 X'09')
              ID(OLD408 NEW508)

For open device:
     CRECOVER DEVN(X'0D08') PRIM(X'FFFF' 32496 X'6A' X'11') SEC(X'FFFF' 32010 X'D4' X'1A')
              OPENDVCS(YES)
```

CSUSPEND – suspending volume pairs

Use the CSUSPEND command to suspend PPRC operations between a primary and secondary volume pair. PPRC stops transferring data to the secondary volume. The primary storage subsystem, however, records all cylinders that change on the primary volume (except when you specify the PRIMARY option, as described below).

You can direct the CSUSPEND command to either the primary or secondary volume of a PPRC volume pair as a result of the DEVN parameter.

Note:

1. The ESS storage subsystem does not accept the CSUSPEND command with the PRIMARY parameter and fails the command if you specify the PRIMARY parameter.
2. Avoid issuing a CDELPAIR command following a CSUSPEND command, as this sequence can cause unpredictable results.

PRIMARY is an optional parameter of the SUSPEND command. The results of using it versus not using it are described as follows:

When you do not specify PRIMARY option —

PPRC suspends the pair. Applications continue to update the primary volume. These updates are not copied to the secondary volume, but the storage control records which cylinders are changed while the pair is suspended. When you reestablish the pair, specify the RESYNC parameter of the CESTPAIR command to direct PPRC to copy only the changed cylinders to the secondary volume. PPRC then returns the secondary volume to duplex state.

This is the default option, and you can address this option to the primary or the secondary volume.

When you specify the PRIMARY option —

PPRC suspends the pair and unit checks all write data I/O that is directed to the device, except for ICKDSF media maintenance channel programs. When ICKDSF media maintenance operations have completed, you can reestablish the pair with the CESTPAIR command with the RESYNC option. (See “Controlling ICKDSF activity to PPRC volumes” on page 283 for a PPRC volume repair procedure.) Only use the PRIMARY option as part of an ICKDSF media maintenance procedure. Ensure that the DEVN parameter specifies the primary volume.

Issue the CESTPAIR command with the RESYNC option to resume PPRC operations for the volume pair.

CSUSPEND command syntax

The syntax of the CSUSPEND command is:

```
►►──CSUSPEND──DEVN──(──device_number──)──┬────────────────────┬──────────────►
                                          └─SUBCHSET(subchset)─┘

►─PRIM──(──ssid serialno cca|lun lss──)───────────────────────────────────────►

                                        (1)
►─SEC──(──ssid serialno cca|lun lss──)─────┬──────────────────────────┬───────►
                                            │              ┌─NO──┐     │
                                            └─OPENDVCS(────┴─YES─┴──)─┘

►─┬─────────┬─────────────────────────────────────────────────────────────────►◄
  └─PRIMARY─┘
```

Notes:

1 *lss* LSS number is required if the storage control supports logical subsystems (like the ESS)

Required parameters

DEVN

Specifies the device number of the primary or secondary volume of the volume pair that PPRC is to suspend. The device number is the 4-digit hexadecimal address of the device to which the I/O operation is directed.

If OPENDVCS(YES) is specified, DEVN identifies a System z device to act as a CKD access volume. This device must be located in the same subsystem cluster as the device (primary or secondary) being suspended.

PRIM Specifies the primary site storage subsystem ID (*ssid*), and serial number (*serialno*), the primary volume channel connection address (*cca*), and the logical storage subsystem (*lss*).

Where:

ssid Specifies a 4-character hexadecimal subsystem identifier (SSID). *ssid* must be specified in X'' notation. When OPENDVCS(YES) is specified, *ssid* can be the actual ssid, or X'FFFF' can be specified and the control unit will calculate the correct FB ssid.

serialno

Specifies the storage control serial number, which can include up to 12 digits, depending on the type of storage control.

cca Specifies a 2-digit channel connection address. *cca* must be specified in X'' notation.

lun Specifies the 2-digit logical unit number when OPENDVCS(YES) is specified. *lun* must be specified in X'' notation.

lss Specifies a 2-digit hexadecimal value that specifies the logical subsystem for the device (ESS only).

Note: LSS number is required if the storage control supports logical subsystems (like the ESS) and not allowed if the storage control does not support logical subsystems.

SEC
Specifies the recovery site storage subsystem ID and serial number, and the secondary volume channel connection address. The descriptions for the values are the same as for the PRIM parameter.

Optional parameters

OPENDVCS
Specifies whether Open System (fixed block) devices or CKD devices are addressed in the required parameters of this command. The values are:

NO
The required parameters address CKD devices. The default is NO.

YES
Specifies that the required parameters in this command target fixed block devices using a CKD access device.

PRIMARY
Directs PPRC to suspend the PPRC volume pair and to reject all write data I/O to the primary volume except for ICKDSF media maintenance channel programs. PRIMARY is valid only when directed to a primary volume with the DEVN parameter. The ESS storage subsystem does not accept the CSUSPEND PRIMARY parameter, and fails the command.

Use the PRIMARY option only as part of an ICKDSF media maintenance procedure because it causes the storage control to unit check all application write I/O.

Note: PRIMARY is not valid with OPENDVCS(YES).

SUBCHSET
Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is supported by the host system's processor and configured for the device.

If you omit the parameter, the command is issued to the device that is logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

CSUSPEND command examples

The following are examples of the CSUSPEND command:

```
For CKD device:
 CSUSPEND DEVN(X'B3C4') PRIM(X'B300' 53114 X'C4' X'03') SEC(X'9500' 58036 X'29' X'05')

For Open Device:
 CSUSPEND DEVN(X'4624') PRIM(X'FFFF' 53114 X'49' X'18') SEC(X'FFFF' 27663 X'32' X'14')
            OPENDVCS(YES)
```

PSETCHAR – set PPRC volume pair characteristics

Use the PSETCHAR command to set characteristics of a PPRC primary and secondary volume pair.

PSETCHAR command syntax

The syntax of the PSETCHAR command is:

```
►►──PSETCHAR──DEVN──(──device_number──)──┬──────────────────────┬──────────────►
                                         └─SUBCHSET(subchset)──┘

►─PRIM──(──serialno cca|lun lss──)──SEC──(──serialno cca|lun lss──)─────────────►

                                        ┌─NO──┐
►──┬──────────────────────────┬──USEFORPM(──┴─YES─┴──)────────────────────────►◄
   │               ┌─NO──┐    │
   └─OPENDVCS(──┴─YES─┴──)─┘
```

Required parameters

DEVN
: Specifies the device number of the primary volume of the volume pair. The device number is the 4-digit hexadecimal address of the device to which the I/O operation is directed.

  If OPENDVCS(YES) is specified, DEVN identifies a System z device to act as a CKD access volume. This device must be located in the same subsystem cluster as the primary device.

PRIM Specifies the primary site storage serial number (*serialno*), the primary volume channel connection address (*cca*), and the logical storage subsystem (*lss*).

Where:

serialno
: Specifies the storage control serial number, which can include up to 12 digits, depending on the type of storage control.

cca Specifies a 2-digit channel connection address.

lun Specifies the 2-digit logical unit number when OPENDVCS(YES) is specified.

lss Specifies a 2-digit hexadecimal value that specifies the logical subsystem for the device.

SEC Specifies the recovery site storage subsystem ID and serial number, the secondary volume channel connection address, and the logical storage subsystem. The descriptions for the values are the same as for the PRIM parameter.

USEFORPM
: Specifies whether the indicated pair is to be used for a Preserve Mirror function in a Multi-Target Mirror configuration. One and only one pair from a PPRC primary should have this enabled or Preserve Mirror Required (Remote Pair FlashCopy) functions will fail due to ambiguous settings. The values are:

NO This pair is not to be used for a Preserve Mirror function in a Multi-Target Mirror configuration.

YES This pair is to be used for a Preserve Mirror function in a Multi-Target Mirror configuration.

Optional parameters

OPENDVCS

Specifies whether Open System (fixed block) devices or CKD devices are addressed in the required parameters of this command. The values are:

NO The required parameters address CKD devices. The default is NO.

YES Specifies that the required parameters in this command target fixed block devices using a CKD access device.

SUBCHSET

Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are the subchannels supported by the system.

If you omit the parameter, the command is issued to the device that is currently logically in subchannel set 0. If a swap has occurred, this could be the device that was physically defined in an alternate subchannel set.

PSETCHAR command examples

The following are examples of the PSETCHAR command:

```
For CKD device:
 PSETCHAR DEVN(0F41) PRIM(CPZ21 00 30) SEC(G3531 00 30) USEFORPM(YES)


For Open Device:
 PSETCHAR DEVN(0F41) PRIM(CPZ21, 00, 50) SEC(G3531, 00, 51) OPENDVCS(YES) USEFORPM(NO)
```

Chapter 15. Managing Peer-to-Peer Remote Copy operations

This section explains the major PPRC system dependencies. It also describes how to identify PPRC volume states, establish PPRC paths, and manage PPRC volumes and paths.

For additional information about disaster recovery procedures, refer to Chapter 16, “Peer-to-Peer Remote Copy data recovery operations,” on page 367.

For additional information about error conditions within the PPRC environment, refer to Chapter 18, “Recovering from Peer-to-Peer Remote Copy error conditions,” on page 381.

In this topic

The following subjects are included in this topic:

Managing the PPRC system

The topics that follow describe how peer-to-peer remote copy interacts with various system components, and how these components affect PPRC.

System IPL volumes

The operating system views PPRC secondary volumes much like dual copy volumes, and permits only a limited set of I/O operations to these volumes. To issue the CRECOVER command during recovery, you must IPL a recovery system that is separate from all PPRC volumes. The CRECOVER command removes all PPRC volumes from secondary PPRC volume status.

Attention: Do *not* allow the data sets that you will require to initialize the recovery system to be PPRC secondary volumes. These volumes include SYSRES, PAGE, and SPOOL volumes, and volumes that are used to startup JES and TSO.

Alternatively, you can have a second set of SYSRES, PAGE, and SPOOL volumes that PPRC keeps current. You would then use a separate set of these volumes for recovery. When recovery is complete, you can use the separate set of SYSRES, PAGE, and SPOOL volumes to initialize the recovered system. Another option is to use ICKDSF Stand-Alone to restore the necessary system data sets.

ESCON Manager

PPRC paths and operations are not known to the ESCON Manager. As a result, you must consider all of the existing physical path connections when you make PPRC changes to the ESCON Director. This includes configuration changes that are made at the ESCON Director console or through the integrated systems management feature of the ESCON Manager, because no protection (using ESCON Manager) is available.

Resource Measurement Facility (RMF) data

Each primary host system write I/O operation passes through the PPRC connection to the recovery site storage control, and then to the recovery system disk devices. RMF statistics do not currently report PPRC connection activity.

RMF reports list reasons for I/O operation delays, including storage control, and ESCON Director port-busy states. Even though PPRC activity may be responsible for these busy conditions, RMF does not explicitly track PPRC activity. PPRC activity may delay primary system I/O operations. You should consider this when you do your capacity planning as well as RMF data interpretation.

DEVSERV and IDCAMS

Use the DEVSERV and IDCAMS functions to determine the status of the storage control's dual copy, cache, and NVS functions. These functions do not report PPRC copy volume pairs or states. The CQUERY TSO command is the only way to obtain information about PPRC volumes.

Identifying Peer-to-Peer Remote Copy volume states

In order to manage peer-to-peer remote copy (PPRC) operations, you need to know the state of PPRC volumes. Figure 21 on page 323 shows the different volume states possible with PPRC. There are additional states, SUSPEND.PM and SUSPEND.CG, not included in the illustration.

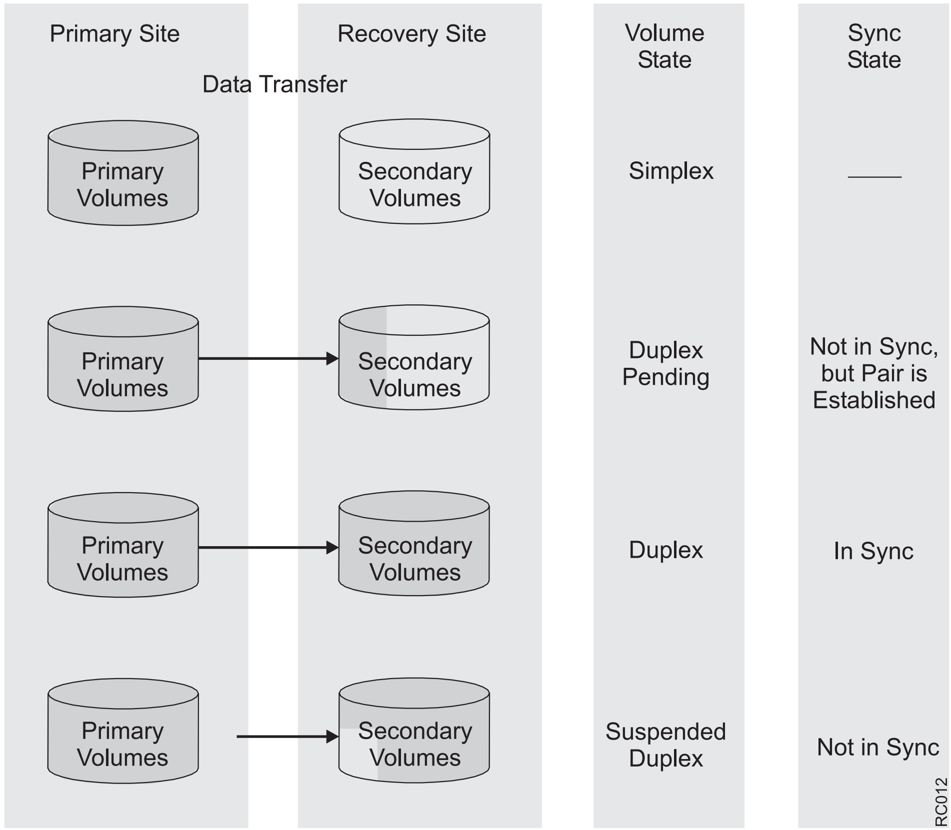


Figure 21. PPRC volume states

To determine the state of a volume, issue the CQUERY command to that volume. "Querying PPRC volumes" on page 330 describes an example of the command and the returned result.

The possible states for a volume are:
- SIMPLEX
- PENDING
- PENDING.XD
- DUPLEX
- SUSPENDED
- SUSP(*n*).XD
- SUSPEND.CG
- SUSPEND.PM
- MTIR
- UNKNOWN

The states that apply to volumes in cascading PPRC relationships are identified as follows:
- SIMPLEX.CAS
- PNDING.CAS
- PNDING.X.C
- DUPLEX.CAS
- SUSP.CAS

- SUSP(*n*).CAS
- SUS(*n*).X.C

The following definitions apply to volume states:

SIMPLEX
: The initial state of a volume.

PENDING
: The initial state of a defined volume pair. This state can also occur after a storage subsystem failure when PPRC reestablishes the volume pair. PPRC is in the process of copying data from the primary volume to the secondary volume.

PENDING.XD
: The state of a volume pair established in PPRC extended distance while the copy operation is in process.

DUPLEX
: The state of a volume pair after PPRC has completed the copy operation, and the volume pair is in synchronization.

MTIR
: The state of a volume in a Multi-Target Internal Relationship (MTIR pair). This is an internal relationship that is created between two multi-target secondary volumes when the Multi-Target Mirror environment is established (when multiple secondaries are created using the PPRC establish command). It is also known as the Multi-Target Incremental Resync relationship.

  If multi-target PPRC is disabled in PARMLIB, the state is UNKNOWN rather than MTIR.

Cascading States
: The state of two volume pairs established through another application (for example, ICKDSF or ESS Specialist) such that the secondary of one pair is the primary of another pair.

SUSPENDED
: The state of a volume pair when the storage subsystems cannot keep the primary and secondary volumes synchronized, or when either the primary host or recovery site host issues a CSUSPEND command. A PPRC volume pair will go into suspended state, for instance, when the primary system storage control fails to complete a write operation to the recovery system storage control. During this suspended state, the primary volume's storage control records the cylinders that applications update. When you issue a CESTPAIR command with the RESYNC parameter, PPRC must only recopy data in the cylinders that have changes. Both volumes return to a synchronized state.

SUSP(*n*). XD
: The state of a volume pair established in PPRC extended distance and in suspended state.

SUSPEND.CG
: The PPRC pair was suspended as a result of a request to pause the Global Mirror session with consistency.

SUSPEND.PM
: The state of a volume pair when a FlashCopy Establish with Preserve Mirror Required was requested and the inband FlashCopy Establish sent from the local control unit failed, resulting in the Local B to Remote B

PPRC pair being suspended. Local B and Remote B are FlashCopy targets. For more information, see "Preserving mirroring" on page 487.

UNKNOWN
The state of a volume in a Multi-Target Internal Relationship (MTIR pair) if multi-target PPRC is disabled in PARMLIB, or the CQUERY is issued with UNFORMAT parameter.

Establishing PPRC paths

Use the CESTPATH command to establish paths between primary site and recovery site logical subsystems or storage controls. A logical subsystem or storage control for a primary volume can have up to eight paths established between it and the logical subsystem or storage control for the pair's secondary volume. Each CESTPATH command can establish up to eight of these paths.

The logical subsystem or storage control for a specific primary (source) volume may, at any one time, link to a maximum of 16 logical subsystems or storage controls for secondary (target) volumes. You must issue a separate CESTPATH command to establish each path between different logical subsystems or storage control. Up to eight links can be specified on one CESTPATH command. See Figure 11 on page 266 for examples of PPRC path options.

Establish paths before issuing CESTPAIR commands. If you intend to establish paths in both directions between the logical subsystem or storage control and host system (as is needed for P/DAS operations), establish each path before you establish the volume pairs.

The primary volume's logical subsystem or storage control keeps information about the paths that connect it to the secondary volume's logical subsystem or storage control. The primary volume's logical subsystem or storage control automatically attempts to restart a failed path, and then resumes copy operations. If all paths to the secondary volume fail, PPRC suspends the pair. You must re-establish these paths with the CESTPATH command after you have corrected the cause of the failed paths.

You can establish paths that use Fibre Channel Protocol (FCP) and paths that use ESCON. You can never mix the two within the same LSS to LSS association. As a reminder, the primary and secondary logical subsystem or storage control must be configured to include the following:
- PPRC-capable Licensed Internal Code (LIC) must be installed on both the primary and secondary logical subsystem or storage control.
- For ESS, the appropriate PPRC feature (V1 or V2) must be enabled.

For additional information about hardware requirements, refer to "PPRC hardware requirements" on page 263.

General overview for establishing a FCP path

Use the PPRC TSO CESTPATH command or the ANTRQST API PESTPATH request to activate up to 8 links between storage subsystems. You can repeat the PPRC CESTPATH command or the ANTRQST API PESTPATH request designating different devices and a different secondary (remote) subsystem up to a total of 16 secondary subsystems. Each subsystem can be connected with up to 8 paths.

Each ESS is assigned a World Wide Node Name (WWNN). The WWNN is used in the CESTPATH command or the PESTPATH request to identify the local and remote ESS subsystems. To establish and use a FCP path you must supply the WWNN identity for both the primary and secondary subsystems.

The WWNN is exactly 16 hexadecimal characters long. Use the PPRC TSO CQUERY command to obtain the WWNN for the primary and secondary subsystems. If you are not able to use the PPRC TSO CQUERY command you can use the ESS Specialist application. The WWNN is on the main (first one that comes up) Specialist screen.

The following is an example of using the CESTPATH command to establish an FCP path:

```
 CESTPATH DEVN(X'A347') PRIM(X'A300' 5005076300C56560 X'03')
SEC(X'D400' 5005076300C63584 X'04')
              LINK(X'00240028') CGROUP(YES)
```

You can display linkage information with this command:

CQUERY DEVN(*devn*) SWWNN(*swwnn*) LINKINFO

The resulting information includes subchannel set, world wide node name and the IDs of primary and secondary linkage adapters.

General overview for establishing ESCON paths

Figure 22 on page 327 shows an example of two separate storage subsystems. In this example, device number 142 is the source volume (the volume that you plan to copy) and becomes the primary volume in the PPRC duplex volume pair. With respect to primary volume 142, the primary system consists of the host processor with ESCON channels, the ESCON Director (ESCD), and Storage Control A. Target volume 262 (the volume to contain the copy) resides on Storage Control B, and becomes the secondary volume in the duplex volume pair. Storage Control B, which may be in the same building or may be many kilometers away, is therefore part of the recovery system for this volume pair.

This installation has established a connection from the host to the primary volume through ESCD ports A6 and A5, and a host-to-secondary connection through ESCD ports A6 and B0.

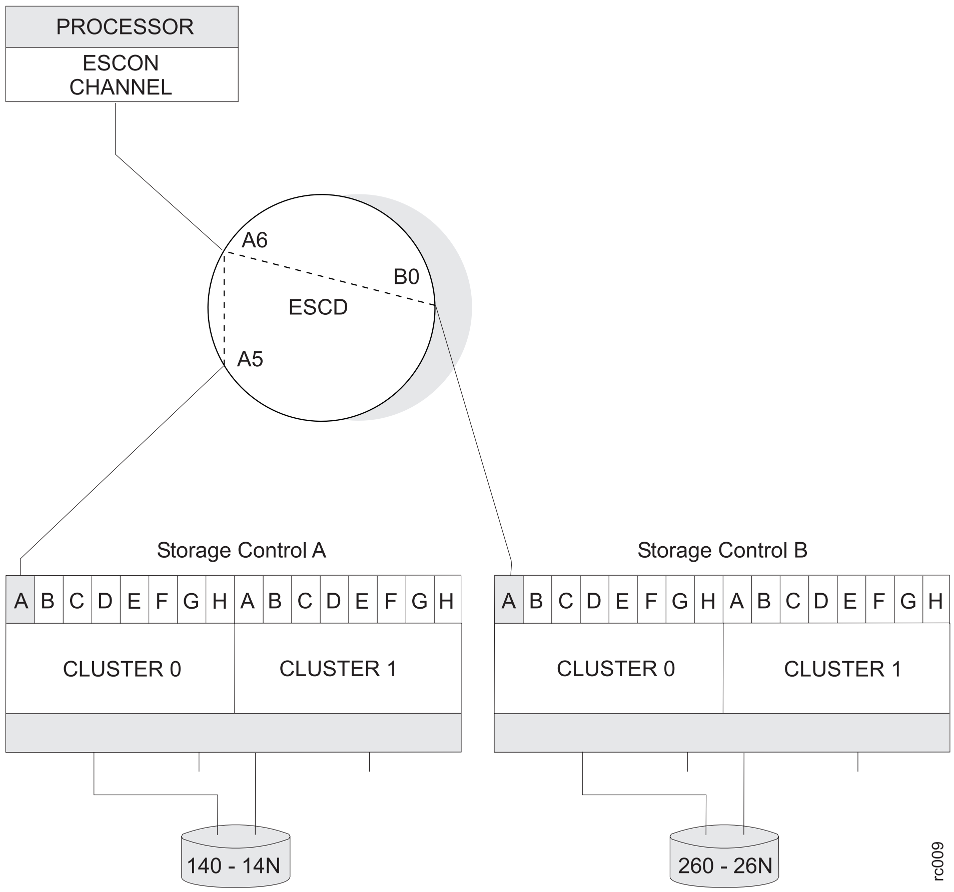


Figure 22. Storage subsystems for primary and secondary volumes

Figure 23 on page 328 shows a PPRC link established between (primary) storage control A and (recovery) storage control B. The CESTPATH command 'LINK(aaaabbcc)' parameter to establish this link would be LINK(0000B000), where the following terms apply:

```
aaaa = primary volume's storage control cluster 0, interface A
  bb = the link destination address, ESCD port B0
  cc = For those storage controls supporting logical subsystems, this value
       is the logical subsystem number. For those storage controls not
       supporting logical subsystems, and for FB LSS, this value is 00.
```

Note:

1. For storage subsystems that support LCUs, see the documentation for that storage subsystem.
2. Before you issue the CESTPATH command, you may want to issue a CQUERY command to a device within each storage subsystem to determine the SSID and the serial number. The CESTPATH command requires this information. You can address the CQUERY command to either a simplex device or a duplex device.
3. You can collect path information with ICKDSF (release 17 or above) using the ANALYZE command, specifying the NOSCAN and NODRIVE keywords. Refer to *Device Support Facilities (ICKDSF) User's Guide and Reference* for more information.

The CESTPATH command for the example in Figure 23 on page 328, using primary volume 0142, is:

```
CESTPATH DEVN(X'0142') PRIM(X'6060' 62019) SEC(X'6061' 68006)
         LINK(X'0000B000')
```

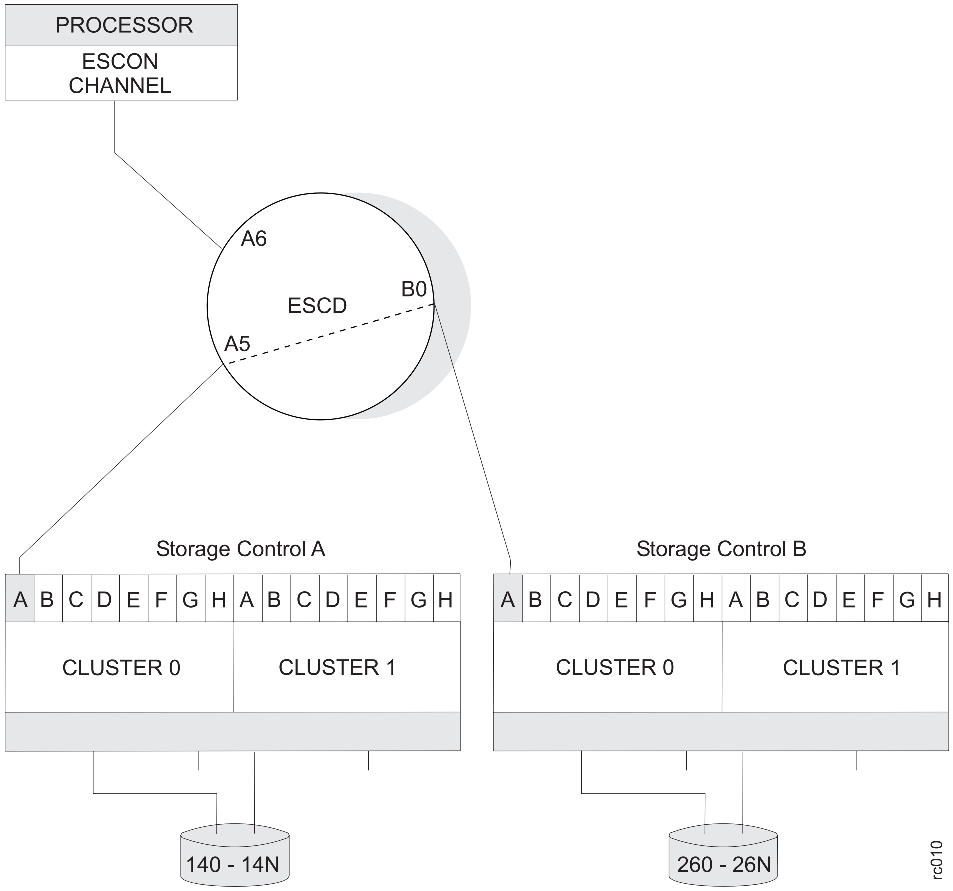


Figure 23. Establishing a PPRC path through an ESCD

Note: The CESTPATH establishes the PPRC link, only. You must also previously configure the ESCD to establish the logical system connection between ports A5 and B0.

You do not need to go through an ESCON Director to establish a PPRC link. Figure 24 on page 329 shows an example of a direct link between storage subsystem A and storage subsystem B. The installation has installed an ESCON fiber optic cable between them.

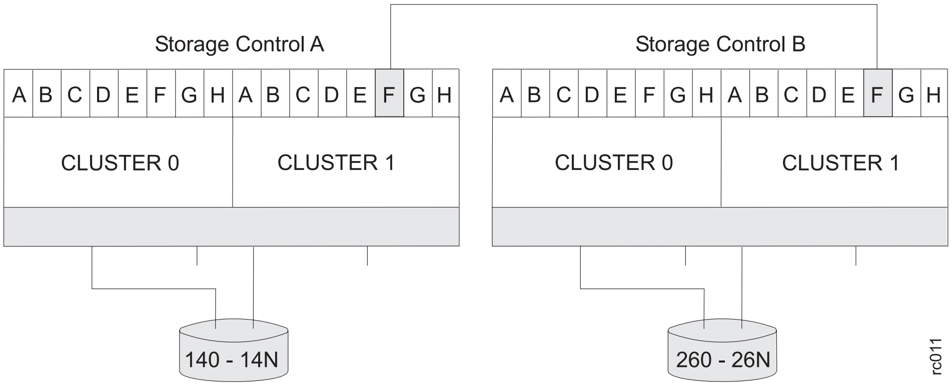


Figure 24. Establishing a direct PPRC path

The 'LINK(aaaabbcc)' parameter of the CESTPATH command for this example would be LINK(00150000), where the following terms apply:

```
aaaa = primary volume's storage control cluster 1, interface F
  bb = 00, as this is a direct connection
  cc = For those storage controls supporting logical subsystems, this value
       is the logical subsystem number. For those storage controls not
       supporting logical subsystems, and for FB LSS, this value is 00.
```

Note:

1. For storage subsystems that support LCUs, see the documentation for that storage subsystem.
2. More information on the CESTPATH command LINK parameter can be found "CESTPATH – establishing paths" on page 304.

The CESTPATH command for the example in Figure 24, using primary volume 0142, is:

```
CESTPATH DEVN(X'0142') PRIM(X'6060' 62019) SEC(X'6061' 68006)
         LINK(X'00150000')
```

Example: The following example adds four ESCON paths between the primary site storage control (SSID 6060, serial number ending with 62019) and the recovery site storage control (SSID 6061, serial number ending with 68006). Issue the CESTPATH command to any device on the primary site storage control.

```
CESTPATH DEVN(X'0142') PRIM(X'6060' 62019) SEC(X'6061' 68006)
         LINK(X'0000B000' X'0001B100' X'0016B200' X'0017B300')
```

Note: The above example assumes that all of the physical and logical paths already exist.

Managing volumes and paths

This section describes how to add, delete, and suspend volume pairs, query a peer-to-peer remote copy volume, and delete a PPRC path.

Adding a PPRC volume pair

Use the CESTPAIR command to specify the PPRC primary and secondary volumes that will make up the pair. Issue the CESTPAIR command to the primary (source) device address. The CESTPAIR command also allows you to add a PPRC volume pair using PPRC extended distance mode and then transition the volume pair to synchronous mode. For information, see "Establishing PPRC volume pairs" on page 355.

Before you issue the CESTPAIR command, you may want to first issue a CQUERY command to both devices that will become the PPRC duplex pair. The returned information includes the storage control SSID, serial number and CCA of the device. The CESTPAIR command requires this information.

Note:

1. To change the value of a CESTPAIR command parameter for an established PPRC volume pair, first issue a CSUSPEND command. Next, reestablish the pair with a CESTPAIR command using the RESYNC parameter specifying all of the desired parameters.
2. If you issue a CESTPAIR command with the RESYNC option to a secondary volume that has been set to simplex with the CRECOVER command, ensure that applications have not modified that volume. If you suspect that the volume data may have changed, reestablish the pair with the CESTPAIR command COPY option. This will protect the data integrity of the two volumes.

The following command adds a synchronous PPRC volume pair:

```
CESTPAIR DEVN(X'0C40') PRIM(X'6060' 62019 X'00')
             SEC(X'6061' 68006 X'09') MODE(COPY) PACE(15) CRIT(N)
```

You can run the following JCL as a batch job, or process it from a PROCLIB to automate the procedures that are shown here. This example shows how to add four PPRC paths and a volume pair to PPRC.

```
//IKJEFT01  JOB,
     //         MSGLEVEL=(1,1),MSGCLASS=A,NOTIFY=user ID
     //STARTRC   EXEC PGM=IKJEFT01
     //STEPLIB DD DSN=authorized.cmdlib,DISP=SHR
     //SYSTSPRT DD SYSOUT=H
     //SYSTSIN  DD *
       CESTPATH                         +
          DEVN(X'0C44')                 +
          PRIM(X'6060' 62019)           +
          SEC(X'6061' 68006)            +
          LINK(X'00008000' X'00108100' X'00018200' X'00118300')
       CESTPAIR                         +
          DEVN(X'0C40')                 +
          PRIM(X'6060' 62019 X'00')     +
          SEC(X'6061' 68006 X'09')      +
          MODE(COPY)                    +
          PACE(2)                       +
          CRIT(NO)
     /*
```

Note: For PPRC extended distance operations, replace PACE(2) and CRIT(YES) with OPTION(XD).

Querying PPRC volumes

The CQUERY command can query the status of one volume from a PPRC volume pair, a volume defined as both primary and secondary in two cascading PPRC pairs (a relationship defined by ICKDSF or ESS Specialist), or collect information about a volume in the simplex state. To do this, either include the VOLUME parameter, or do not specify any optional parameters. The following command queries volume 0F40:

```
CQUERY DEVN(X'0F40')
```

The path information that is available for the storage control depends on the current status of the volume to which the CQUERY command is directed, as follows:

- If the device is in **simplex** state, PPRC may display path information for the logical subsystem or the storage control for the device. The path information does not relate to the specified device.
- If the device is a **primary** of a PPRC volume pair, PPRC displays path information for the logical subsystem or the storage control. The path information does relate to the specified device. The devices that make up the volume pair are using one or more paths.
- If the device is a **secondary** of a PPRC volume pair, PPRC provides a report, but valid path information is not available. When you issue a CQUERY PATHS command to an active PPRC pair secondary volume, the formatted CQUERY report will not display the serial number or path link information. The WWNN is displayed, if available.

In a Multi-Target Mirror environment, that is, Multi-Target is supported by the microcode and ENABLE(PPRCMT) is specified in the DEVSUP*xx* member of PARMLIB, the output is formatted into multiple VOLUME REPORTs within an ANTP0090I message block, just as is done when you query the middle device of a cascading relationship.

Formatted and unformatted output

When you issue the CQUERY command for a volume, you can request:

Formatted output
: This returns status as a block message to your TSO session or batch JOBLOG, and to the SYSLOG. Request formatted output with the FORMAT parameter (the default).

Unformatted output
: This allows you to save screen space by returning unformatted status information. Request unformatted output with the UNFORMAT parameter.

Note: CQUERY command output in the SYSLOG can contain a connection code at the end of the header message and at the beginning of each of the lines associated with that connection code. The system generates this code for any multiline WTO.

Examples of formatted and unformatted output follow. For a description of the fields in formatted output, refer to “Fields in formatted output” on page 332. For a description of the information in unformatted output, refer to message ANTP0091I in *z/OS MVS System Messages, Vol 1 (ABA-AOM)*.

Example of formatted output

The following output is displayed for a CQUERY FORMAT command for a primary volume.

```
15.12.00 TSU00189 ANTP0090I CQUERY FORMATTED LVL 3            C
VOLUME REPORT
*********** PPRC REMOTE COPY CQUERY - VOLUME *****************
*                                    (PRIMARY)   (SECONDARY) *
*                                   SSID CCA LSS SSID CCA LSS*
*DEVICE   LEVEL   STATE    PATH STATUS  SERIAL#   SERIAL#    *
*------ -------  --------  ----------- ---------  ---------  *
* 0F40  PRIMARY.. DUPLEX..  ACTIVE..    0240 03    8080 01   *
* SCH(0) CRIT(NO).......  CGRPLB(NO).  0000090383 00000090461*
* PATHS SAID/DEST STATUS: DESCRIPTION                        *
* ----- --------- ------  -------------------                *
*   1   0011 0000   01    PATH ESTABLISHED...                *
*   2   0012 0000   01    PATH ESTABLISHED...                *
*   3   0013 0000   01    PATH ESTABLISHED...                *
*   4   0014 0000   01    PATH ESTABLISHED...                *
*   5   0015 0000   01    PATH ESTABLISHED...                *
*   6   0016 0000   01    PATH ESTABLISHED...                *
*   7   ---- ----   00    NO PATH............                *
*   8   ---- ----   00    NO PATH............                *
**************************************************************
ANTP0001I CQUERY COMMAND COMPLETED FOR DEVICE 0F40.
 COMPLETION CODE: 00
```

Fields in formatted output

Formatted output for the CQUERY command contains the following fields:

DEVICE
: Displays the 4-digit hexadecimal device address. SCH(*x*) indicates the subchannel for the device.

LEVEL
: Displays whether the volume is a primary or secondary volume. For a primary volume, it also displays the status of the volume as defined by the CRIT parameter of the CESTPAIR command that established the pair.

If neither PRIMARY nor SECONDARY is displayed, the volume is not part of a PPRC pair.

CRIT(NO) indicates that you either issued the CESTPAIR command without the CRIT parameter, or specified CRIT(NO).

CRIT(YES-PATHS) indicates that the volume pair only be write-inhibited when there are no PPRC paths available. CRIT(YES-ALL) indicates that the primary storage control has been configured to suspend the pair and prevent I/O to the primary volume as a result of any error, including an error on the primary or secondary volume. CRIT(YES-PATHS) and CRIT(YES-ALL) both indicate that the CESTPAIR command included the CRIT(YES) parameter.

STATE
: Indicates whether the PPRC volume is in simplex, pending, duplex, suspended (suspend(*n*)), extended distance pending XD, Multi-Target Internal Relationship (MTIR) or extended distance susp(*n*).xd state. The value *n* for SUSPEND is one of the following:

- (3) — The PPRC volume pair was suspended by a host command to the primary site storage control.
- (4) — The PPRC volume pair was suspended by a host command to the recovery site storage control.

- (5) — A command issued by the primary site storage control to the recovery site storage control has suspended the pair. Only a recovery site storage control returns value 5.
- (6) — Internal conditions within either storage control have suspended the pair. Either storage control returns value 6.
- (7) — The transition of the recovery volume to the simplex state has suspended the pair. Only a primary site storage control returns value 7.
- (8) — The primary site storage control has suspended the pair as a result of abnormal conditions within the recovery site storage subsystem. These conditions may involve the storage control, its attached disk devices, and the ESCON paths between the two sites. The primary site storage control detects and reports this condition.
- (9) — Either an IPL, or a power interruption to either storage control, have suspended the pair. The storage control that received the interruption returns value 9.
- (A) — A CGROUP command with the FREEZE parameter has suspended the pair. You must query a primary device to receive value A.

PATH STATUS
Displays whether the status is active or inactive.

SSID CCA LSS Serial # PRIMARY
Displays the four-character subsystem ID (SSID) of the primary site storage control, the two-character LSS CCA of the primary volume and, when available, the storage control serial number (up to 12 characters).

SSID CCA LSS Serial # SECONDARY
Displays the four-character subsystem ID (SSID) of the recovery site storage control, the two-character LSS CCA of the primary volume, and the storage control serial number (up to 12 characters).

CRIT Displays the status of the CRIT keyword of the CESTPAIR command.

CGRPLB
Indicates whether or not the storage subsystem is configured to support PPRC consistency grouping.

INCRES
Indicates whether the incremental resync change recording mechanism has been started for the Metro/Global Mirror environment.

Additional status lines for the queried device might be displayed:

```
THIS DEVICE IS CURRENTLY IN A CASCADED, FAILOVER STATE.
THIS DEVICE IS CURRENTLY PRIMED FOR RESYNCHRONIZATION.
PAIR WAS THE TARGET OF A WITHDRAWN PRESERVE MIRROR RELATION.
THIS DEVICE IS CURRENTLY IN A SOFT FENCE STATE.
THIS PAIR HAS BEEN IDENTIFIED TO BE USED FOR PRESERVE MIRROR.
```

PATHS
Displays the number of valid links, if there are any, between primary site and recovery site storage controls.

SAID Displays the four-character system adapter ID (SAID) of the primary site storage control. SAID and DEST (described below) make up the link address. If there are no paths, the SAID/DEST header is the default.

DEST Specifies up to four 4-digit, hexadecimal-character link addresses. The link addresses each consist of a 2-digit, hexadecimal-character ESCON port address followed by 00.

XXXX XXXX --
The storage control has an association recorded between the indicated primary and secondary subsystems, but 0 paths identified, so was it not able to determine the SAID and DEST. A path and a status of FF will be displayed. This may occur after a freeze is issued. To clean up the "XXXX XXXX" values, an establish path must be issued. The SSID, CCA, LSS, Serial number, and WWNN are given to help determine the correct primary and secondary parameters to use for the establish path command. If you are continuing operations after a freeze, reestablish the path and pairs. If a link between these two subsystems is no longer required, the association should be removed by reestablishing the path between the primary and secondary subsystems, deleting any suspended pairs, and then deleting the path.

PFCA Displays the primary Fibre Channel adapter identifier.

SFCA Displays the secondary Fibre Channel adapter identifier.

STATUS
Displays a 2-digit number that indicates the detailed status of the path.

DESCRIPTION
Provides a short description of the status indicator:

- 00 — No path
- 01 — Established path
- 02 — Either initialization or reinitialization has failed, or the path is inoperative
- 03 — Time out
- 04 — No resources available at primary site storage control
- 05 — No resources available at recovery site storage control
- 06 — Serial number mismatch
- 10 — Configuration error
- 13 — Fibre Path Established
- 14 — Fibre Channel Path Link Down
- 15 — Fibre Channel Path Retry Exceeded
- 16 — Fibre Channel Path Secondary Adapter not PPRC Capable
- 17 — Fibre Channel Path Secondary Adapter not Available
- 18 — Fibre Channel Path Primary Login Exceeded
- 19 — Fibre Channel Path Secondary Login Exceeded
- 1A — Primary Fibre Channel not Configured
- 1B — Fibre Degraded
- 1C — Fibre Removed
- nn — Status is undetermined
- FF — Unable to determine

FIRST CYL OUT OF SYNCH (displayed for RAMAC Virtual Array (RVA and IBM 3990)
This information, when available, is presented only for volumes in pending and suspended states.

If the volume is a primary volume, this is the lowest cylinder number that is out of synchronization on the primary volume. For a secondary volume, this is the lower cylinder number that was received from the primary. This update is in cache, as PPRC has not yet written this update to the secondary volume.

LAST CYL OUT OF SYNCH (displayed for RAMAC Virtual Array (RVA and IBM 3990)
This information, when available, is presented only for volumes in pending and suspended states.

If the volume is a primary volume, this is the highest cylinder number out of synchronization on the primary volume. For a secondary volume, this is the higher cylinder number that was received from the primary. This update is in cache, as PPRC has not yet written this update to the secondary volume.

TRACKS OUT OF SYNC (displayed for ESSs)
This information displays for volumes that are in PENDING or SUSPEND states. (The ESS must have the most current level of microcode installed.) If the volume is a primary volume, this is the number out-of-sync tracks that PPRC has not yet written to the secondary volume.

TRACKS ON VOLUMES (displayed for ESSs)
This information displays for volumes that are in PENDING or SUSPEND states. (The ESS must have the most current level of microcode installed.) If the volume is a primary volume, this is the number of tracks that exist on the volume.

PERCENT OF COPY COMPLETED
Displays the percent completed for the copy operation based on the number of cylinders or tracks on the particular type of storage control. This is an optional line of information that PPRC presents only if the volume is a primary volume and the information is available.

SECONDARY WAS SUSPENDED
When it is available, displays the Universal Time Coordinated (UTC) to indicate when this secondary volume became suspended. The UTC, in the ISO format “YYYY-MM-DD HH.MM.SS.NNNNNN”, is the time-of-day clock value from the host system from which the CSUSPEND command was issued.

SUBSYSTEM
Indicates all the primary and secondary storage controls that are associated with each WWNN that may have been established by a CESTPATH request. The presence of either PRIMARY or SECONDARY indicates that the subsystem is associated with a PPRC path.

The subsystem may have PPRC activity that is unrelated to the device for which the query was issued. The WWNN information may be included in the CQUERY 'VOLUME REPORT' as long as there is an active Fibre Channel Protocol (FCP) path on the device's associated LSS. The device for which the query was issued may not show active paths on the CQUERY 'VOLUME REPORT'. If the CQUERY 'PATHS REPORT' indicates active paths for the device, the WWNN information may be related to that device.

PRIMARY indicates that the associated WWNN is for a primary (or source) logical subsystem or storage control. SECONDARY indicates that the associated WWNN is for a secondary (or target) logical subsystem or storage control.

The absense of either PRIMARY or SECONDARY next to a WWNN indicates that the query request could not determine the status of the subsystem. The WWNN is provided as informational data.

WWNN (Primary, Secondary)
Specifies the World Wide Node Name (WWNN) that is used to access a primary or secondary logical subsystem or storage control referenced by the SSID using fibre channel protocol (FCP) linkage. Every ESS is assigned a WWNN and is 16 hexadecimal digits in length.

The presence of the WWNN information indicates that there is one or more paths established in the LSS associated with the device for which the QUERY was issued.

UNKNOWN — means the Query was not able to determine the WWNN of a subsystem.

Note: The SUBSYSTEM and WWNN values are only displayed when a logical subsystem or storage control is FCP enabled.

- For additional information about PPRC volume states, see "Identifying Peer-to-Peer Remote Copy volume states" on page 322.
- For additional information about the CRIT parameter, see "CESTPAIR – establishing volume pairs" on page 296.
- For additional information about PPRC status descriptions for paths, see "Querying PPRC paths" on page 345.

Examples of CQUERY output

Examples of formatted and unformatted output follow.

CQUERY formatted output for a common volume in two cascading PPRC pairs: The following output displays for a volume that is primary and secondary in two cascading PPRC pairs defined by ICKDSF or ESS Specialist.

```
ANTP0090I CQUERY FORMATTED LVL 3
VOLUME REPORT
************** PPRC REMOTE COPY CQUERY - VOLUME ********************
*                                          (PRIMARY)   (SECONDARY) *
*                                         SSID CCA LSS SSID CCA LSS*
*DEVICE   LEVEL      STATE     PATH STATUS  SERIAL#     SERIAL#    *
*------ ---------  ----------  ----------- ---------    ---------  *
* 0F70  SECONDARY  DUPLEX_CAS.  ACTIVE..  8800 3F 00   9800 3F 00 *
* SCH(0) CRIT(NO).......      CGRPLB(NO). 000000050706 000000050344*
* PATHS SAID/DEST STATUS: DESCRIPTION                              *
* ----- --------- ------  -------------------                      *
*   8   2008 0800   01    PATH ESTABLISHED...                      *
*       2009 0900   01    PATH ESTABLISHED...                      *
*       2018 1800   01    PATH ESTABLISHED...                      *
*       2019 1900   01    PATH ESTABLISHED...                      *
*       200C 0C00   01    PATH ESTABLISHED...                      *
*       200D 0D00   01    PATH ESTABLISHED...                      *
*       201C 1C00   01    PATH ESTABLISHED...                      *
*       201D 1D00   01    PATH ESTABLISHED...                      *
********************************************************************
VOLUME REPORT
************** PPRC REMOTE COPY CQUERY - VOLUME ********************
*                                          (PRIMARY)   (SECONDARY) *
*                                         SSID CCA LSS SSID CCA LSS*
*DEVICE   LEVEL      STATE     PATH STATUS  SERIAL#     SERIAL#    *
*------ ---------  ----------  ----------- ---------    ---------  *
* 0F70  PRIMARY..  DUPLEX_CAS.   ACTIVE..   9800 3F 00   9700 2F 01*
* SCH(0) CRIT(NO).......      CGRPLB(NO). 000000050344 000000042800*
* PATHS SAID/DEST STATUS: DESCRIPTION                              *
* ----- --------- ------  -------------------                      *
*   8   2008 0800   01    PATH ESTABLISHED...                      *
*       2009 0900   01    PATH ESTABLISHED...                      *
*       2018 1800   01    PATH ESTABLISHED...                      *
*       2019 1900   01    PATH ESTABLISHED...                      *
*       200C 0C00   01    PATH ESTABLISHED...                      *
*       200D 0D00   01    PATH ESTABLISHED...                      *
*       201C 1C00   01    PATH ESTABLISHED...                      *
*       201D 1D00   01    PATH ESTABLISHED...                      *
********************************************************************
ANTP0001I CQUERY COMMAND COMPLETED FOR DEVICE 0F70. COMPLETION CODE: 00
```

CQUERY formatted output for a PPRC pair that was previously the target of a Preserve Mirror relationship: When a PPRC pair was previously the target of a Preserve Mirror relationship that was withdrawn, an indicator is present in the PPRC query data, so that the results of a CQUERY command include the text `PAIR WAS THE TARGET OF A WITHDRAWN PRESERVE MIRROR RELATION`, as shown in the following example:

```
ANTP0090I CQUERY FORMATTED LVL 4 174
VOLUME REPORT
************** PPRC REMOTE COPY CQUERY - VOLUME ********************
*                                         (PRIMARY)   (SECONDARY) *
*                                        SSID CCA LSS SSID CCA LSS*
*DEVICE   LEVEL      STATE     PATH STATUS  SERIAL#     SERIAL#    *
*------- ---------  ----------  ----------- ---------    ---------  *
* 0F44  SECONDARY  DUPLEX....   ACTIVE..   C800 3E 00   C800 3F 00 *
* SCH(0) ...............      ........... 000000001691 000000001691*
*      ...........          ...............                       *
*PAIR WAS THE TARGET OF A WITHDRAWN PRESERVE MIRROR RELATION       *
* PATHS PFCA SFCA STATUS: DESCRIPTION                              *
* ----- --------- ------  -------------------                      *
*   1   0200 0202   13    PATH ESTABLISHED...                      *
*       ---- ----   00    NO PATH............                      *
*       ---- ----   00    NO PATH............                      *
*       ---- ----   00    NO PATH............                      *
*  SUBSYSTEM         WWNN                  LIC LEVEL               *
* -----------  ----------------           -----------              *
* PRIMARY....  5005076303FFC05F           5.4.2.356                *
* SECONDARY.1  5005076303FFC05F                                    *
********************************************************************
```

CQUERY unformatted output for a primary volume: The following output is for a CQUERY UNFORMAT command for a primary volume. See message ANTP0091I for details about the reported fields.

```
ANTP8802I CQUERY devn(x'f47') unformat
ANTP0091I CQUERY UNFORMAT LVL 3 275
VOLUME REPORT
0F47,PRIMARY,DUPLEX,ACTIVE,
E2EC0B,0A,0000000BWGV1,E6EC0B,0A,0000000BWHW1,N,N,
1,01000200,13,,,,,,,
,,,
5005076304FFC441,5005076304FFC442,
,,
ANTP0001I CQUERY COMMAND COMPLETED FOR DEVICE 0F47. COMPLETION CODE: 00
```

CQUERY formatted output for a primary volume on an ESS enabled for FCP: The following output displays for a CQUERY FORMAT request for a primary volume on an ESS connected by FCP.

```
 ************** PPRC REMOTE COPY CQUERY - VOLUME ********************
*                                         (PRIMARY)   (SECONDARY) *
*                                        SSID CCA LSS SSID CCA LSS*
*DEVICE   LEVEL      STATE     PATH STATUS  SERIAL#     SERIAL#    *
*------- ---------  ----------  ----------- ---------    ---------  *
* 0F74  PRIMARY..  DUPLEX....   ACTIVE..   C000 30 00   B000 30 00 *
* SCH(0) CRIT(NO).......      CGRPLB(NO). 000000026438 000000026437*
* PATHS PFCA SFCA STATUS: DESCRIPTION                              *
* ----- --------- ------  -------------------                      *
*   1   0028 0024   13    PATH ESTABLISHED...                      *
*       ---- ----   00    NO PATH............                      *
*       ---- ----   00    NO PATH............                      *
*       ---- ----   00    NO PATH............                      *
*  SUBSYSTEM         WWNN                                          *
* -----------  ----------------                                    *
* PRIMARY....  5005076300C0A4DE                                    *
* SECONDARY..  5005076300C0A4DD                                    *
********************************************************************
ANTP0001I CQUERY COMMAND COMPLETED FOR DEVICE 0F74. COMPLETION CODE: 00
```

The following output displays for a CQUERY FORMAT request for a primary volume on an ESS connected by FCP. The WWNN cannot be determined.

```
 ************** PPRC REMOTE COPY CQUERY - VOLUME ********************
*                                         (PRIMARY)   (SECONDARY) *
*                                        SSID CCA LSS SSID CCA LSS*
*DEVICE   LEVEL      STATE     PATH STATUS  SERIAL#     SERIAL#    *
*------ ---------  ----------  ----------- ---------    ---------  *
* 0F74  .........  SIMPLEX...   INACTIVE   C000 30 00   .......... *
* SCH(0) ...............      ........... 000000026438 ............*
* PATHS SAID DEST STATUS: DESCRIPTION                              *
* ----- --------- ------  -------------------                      *
*   0   ---- ----   00    NO PATH...........                       *
*       ---- ----   00    NO PATH...........                       *
*       ---- ----   00    NO PATH...........                       *
*       ---- ----   00    NO PATH...........                       *
*  SUBSYSTEM         WWNN                                          *
* -----------  ----------------                                    *
* PRIMARY....  ..UNKNOWN..                                    *
*******************************************************************
ANTP0001I CQUERY COMMAND COMPLETED FOR DEVICE 0F74. COMPLETION CODE: 00
```

CQUERY formatted output for a Metro/Global Mirror environment in a recovery scenario: The following outputs display for a CQUERY FORMAT request for a Metro/Global Mirror environment in a recovery scenario.

When the volume is in a cascading Failover state, the following reports demonstrate the status.

```
ANTP0090I CQUERY FORMATTED LVL 4 995
************** PPRC REMOTE COPY CQUERY - VOLUME ********************
*                                         (PRIMARY)   (SECONDARY) *
*                                        SSID CCA LSS SSID CCA LSS*
*DEVICE   LEVEL      STATE     PATH STATUS  SERIAL#     SERIAL#    *
*------ ---------  ----------  ----------- ---------    ---------  *
* BD00  SECONDARY  SUSP(4)...   ACTIVE..   1360 00 60   2BE0 00 E0 *
* SCH(0) ...............      ........... 0000000AZFP1 0000000BHYR1*
*       ...........                                                *
*THIS DEVICE IS CURRENTLY IN A CASCADED, FAILOVER STATE.           *
* PATHS PFCA SFCA STATUS: DESCRIPTION                              *
* ----- --------- ------  -------------------                      *
*   1   0032 00A4   13    PATH ESTABLISHED...                      *
*       ---- ----   00    NO PATH...........                       *
*       ---- ----   00    NO PATH...........                       *
*       ---- ----   00    NO PATH...........                       *
*  SUBSYSTEM         WWNN                  LIC LEVEL               *
* -----------  ----------------           -----------              *
* PRIMARY....  5005076304FFC0A7           5.1.600.208              *
* SECONDARY.1  5005076304FFC2F4                                    *
*******************************************************************
```

```
ANTP0090I CQUERY FORMATTED LVL 4 995
************** PPRC REMOTE COPY CQUERY - VOLUME ********************
*                                         (PRIMARY)   (SECONDARY) *
*                                        SSID CCA LSS SSID CCA LSS*
*DEVICE   LEVEL      STATE     PATH STATUS  SERIAL#     SERIAL#    *
*------ ---------  ----------  ----------- ---------    ---------  *
* BD00  PRIMARY..  PENDING.XD   ACTIVE..   2BE0 00 E0   0123 18 03 *
* SCH(0) CRIT(NO).......      CGRPLB(NO). 0000000BHYR1 000000027449*
*       INCRES(NO).                                                *
*THIS DEVICE IS CURRENTLY IN A CASCADED, FAILOVER STATE.           *
* PATHS PFCA SFCA STATUS: DESCRIPTION                              *
* ----- --------- ------  -------------------                      *
*   1   0032 00A4   13    PATH ESTABLISHED...                      *
*       ---- ----   00    NO PATH............                      *
*       ---- ----   00    NO PATH............                      *
*       ---- ----   00    NO PATH............                      *
*                              PERCENT OF COPY COMPLETE = 100%     *
*  SUBSYSTEM         WWNN                  LIC LEVEL               *
* -----------  ----------------           -----------              *
* PRIMARY....  5005076304FFC2F4           5.1.600.208              *
* SECONDARY.1  5005076300C0A8D1                                    *
********************************************************************
```

When the cascaded failover and primed for re-synchronization are both set, the following reports demonstrate the status.

```
ANTP0090I CQUERY FORMATTED LVL 4 995
************** PPRC REMOTE COPY CQUERY - VOLUME ********************
*                                         (PRIMARY)   (SECONDARY) *
*                                        SSID CCA LSS SSID CCA LSS*
*DEVICE   LEVEL      STATE     PATH STATUS  SERIAL#     SERIAL#    *
*------ ---------  ----------  ----------- ---------    ---------  *
* BD00  SECONDARY  SUSP(4)...   ACTIVE..   1360 00 60   2BE0 00 E0 *
* SCH(0) ..............      ........... 0000000AZFP1 0000000BHYR1*
*       ..........                                                 *
*THIS DEVICE IS CURRENTLY IN A CASCADED, FAILOVER STATE.           *
*THIS DEVICE IS CURRENTLY PRIMED FOR RESYNCHRONIZATION.            *
* PATHS PFCA SFCA STATUS: DESCRIPTION                              *
* ----- --------- ------  -------------------                      *
*   1   0032 00A4   13    PATH ESTABLISHED...                      *
*       ---- ----   00    NO PATH............                      *
*       ---- ----   00    NO PATH............                      *
*       ---- ----   00    NO PATH............                      *
*  SUBSYSTEM         WWNN                  LIC LEVEL               *
* -----------  ----------------           -----------              *
* PRIMARY....  5005076304FFC0A7           5.1.600.208              *
* SECONDARY.1  5005076304FFC2F4                                    *
********************************************************************
```

```
ANTP0090I CQUERY FORMATTED LVL 4 995
************** PPRC REMOTE COPY CQUERY - VOLUME ********************
*                                           (PRIMARY)   (SECONDARY) *
*                                          SSID CCA LSS SSID CCA LSS*
*DEVICE   LEVEL      STATE     PATH STATUS  SERIAL#     SERIAL#    *
*------ ---------  ----------  ----------- ---------    ---------  *
* BD00  PRIMARY..  PENDING.XD   ACTIVE..   2BE0 00 E0   0123 18 03 *
* SCH(0) CRIT(NO).......      CGRPLB(NO). 0000000BHYR1 000000027449*
*      INCRES(NO).                                                 *
*THIS DEVICE IS CURRENTLY IN A CASCADED, FAILOVER STATE.           *
*THIS DEVICE IS CURRENTLY PRIMED FOR RESYNCHRONIZATION.            *
* PATHS PFCA SFCA STATUS: DESCRIPTION                              *
* ----- --------- ------  -------------------                      *
*   1   0032 00A4   13    PATH ESTABLISHED...                      *
*       ---- ----   00    NO PATH............                      *
*       ---- ----   00    NO PATH............                      *
*       ---- ----   00    NO PATH............                      *
*                              PERCENT OF COPY COMPLETE = 100%     *
*  SUBSYSTEM         WWNN                 LIC LEVEL                *
* -----------  ----------------           -----------              *
* PRIMARY....  5005076304FFC2F4           5.1.600.208              *
* SECONDARY.1  5005076300C0A8D1                                    *
********************************************************************
```

CQUERY formatted output in a multi-target configuration: This topic includes several examples of CQUERY formatted output in a multi-target configuration.

1. When you query the primary in a multi-target configuration, with multi-target capability enabled, the output includes a volume report for each relationship from the primary.

```
CQUERY FORMATTED LVL 6
VOLUME REPORT
************** PPRC REMOTE COPY CQUERY - VOLUME ********************
*                                           (PRIMARY)   (SECONDARY) *
*                                          SSID CCA LSS SSID CCA LSS*
*DEVICE   LEVEL      STATE     PATH STATUS  SERIAL#     SERIAL#    *
*------ ---------  ----------  ----------- ---------    ---------  *
* 0F90  PRIMARY..  DUPLEX....   ACTIVE..   D000 06 30   D000 07 30 *
* SCH(0) CRIT(NO).......      CGRPLB(YES) 0000000CCH81 0000000CCH81*
*       INCRES(NO).           AUTORESYNC(NO).                      *
* PATHS PFCA SFCA STATUS: DESCRIPTION                              *
* ----- --------- ------  -------------------                      *
*   1   0232 0233   13    PATH ESTABLISHED...                      *
*       ---- ----   00    NO PATH............                      *
*       ---- ----   00    NO PATH............                      *
*       ---- ----   00    NO PATH............                      *
*  SUBSYSTEM         WWNN                  LIC LEVEL               *
* -----------  ----------------           -----------              *
* PRIMARY....  5005076305FFD2A9           7.7.27.12                *
* SECONDARY.1  5005076305FFD2A9                                    *
* SECONDARY.2  5005076305FFD2A9                                    *
********************************************************************
VOLUME REPORT
************** PPRC REMOTE COPY CQUERY - VOLUME ********************
*                                           (PRIMARY)   (SECONDARY) *
*                                          SSID CCA LSS SSID CCA LSS*
*DEVICE   LEVEL      STATE     PATH STATUS  SERIAL#     SERIAL#    *
*------ ---------  ----------  ----------- ---------    ---------  *
* 0F90  PRIMARY..  DUPLEX....   ACTIVE..   D000 06 30   D000 08 30 *
* SCH(0) CRIT(NO).......      CGRPLB(YES) 0000000CCH81 0000000CCH81*
*       INCRES(NO).           AUTORESYNC(NO).                      *
* PATHS PFCA SFCA STATUS: DESCRIPTION                              *
* ----- --------- ------  -------------------                      *
*   1   0232 0233   13    PATH ESTABLISHED...                      *
*       ---- ----   00    NO PATH............                      *
*       ---- ----   00    NO PATH............                      *
*       ---- ----   00    NO PATH............                      *
*  SUBSYSTEM         WWNN                  LIC LEVEL               *
* -----------  ----------------           -----------              *
* PRIMARY....  5005076305FFD2A9           7.7.27.12                *
* SECONDARY.1  5005076305FFD2A9                                    *
* SECONDARY.2  5005076305FFD2A9                                    *
********************************************************************
CQUERY COMMAND COMPLETED FOR DEVICE 0F90. COMPLETION CODE: 00
```

2. When you query the primary in a multi-target configuration from a system that doesn't have multi-target enabled, the output includes the following:

   ```
   ADDITIONAL MULTI-TARGET RELATIONSHIPS EXIST
   ```

   and

   ```
   COMPLETION CODE: 04
   ```

3. When you query the primary in a multi-target configuration with multi-target enabled, and the PSETCHAR command has identified the pair to be used for Preserve Mirror, the output includes this additional information:

   ```
   *THIS PAIR HAS BEEN IDENTIFIED TO BE USED FOR PRESERVE MIRROR.
   ```

   as shown in this excerpt:

   ```
   ANTP0090I CQUERY FORMATTED LVL 6
   VOLUME REPORT
   ************** PPRC REMOTE COPY CQUERY - VOLUME ********************
   *                                           (PRIMARY)   (SECONDARY) *
   *                                          SSID CCA LSS SSID CCA LSS*
   *DEVICE   LEVEL      STATE     PATH STATUS  SERIAL#     SERIAL#    *
   *------ ---------  ----------  ----------- ---------    ---------  *
   * 0F51  PRIMARY..  DUPLEX....   ACTIVE..   D400 00 30   D000 00 30 *
   * SCH(0) CRIT(NO).......      CGRPLB(YES) 0000000CRB41 0000000CRB31*
   ```

```
*       INCRES(NO).           AUTORESYNC(NO).                     *
*THIS PAIR HAS BEEN IDENTIFIED TO BE USED FOR PRESERVE MIRROR.     *
* PATHS PFCA SFCA STATUS: DESCRIPTION                             *
* ----- --------- ------  -------------------                     *
.
.
.
```

4. The following example shows two volume reports for secondaries. In the second, the state is UNKNOWN because multi-target is disabled.

```
ANTP0090I CQUERY FORMATTED LVL 6
VOLUME REPORT
************** PPRC REMOTE COPY CQUERY - VOLUME ********************
*                                          (PRIMARY)   (SECONDARY) *
*                                         SSID CCA LSS SSID CCA LSS*
*DEVICE   LEVEL      STATE     PATH STATUS  SERIAL#     SERIAL#    *
*------ ---------  ----------  ----------- ---------    ---------  *
* 0F61  SECONDARY  DUPLEX.CAS   ACTIVE..   D400 00 30   D812 00 31 *
* SCH(0) ..............      ........... 0000000CRB41 0000000CRB71*
*       ..........        ..............                          *
* PATHS PFCA SFCA STATUS: DESCRIPTION                             *
* ----- --------- ------  -------------------                     *
*   1   0133 0232   13    PATH ESTABLISHED...                     *
*       ---- ----   00    NO PATH...........                      *
*       ---- ----   00    NO PATH...........                      *
*       ---- ----   00    NO PATH...........                      *
*  SUBSYSTEM         WWNN                  LIC LEVEL              *
* -----------  ----------------          -----------              *
* PRIMARY....  5005076305FFD70E          7.7.40.291               *
* SECONDARY.1  5005076305FFD710                                   *
*******************************************************************
VOLUME REPORT
************** PPRC REMOTE COPY CQUERY - VOLUME ********************
*                                          (PRIMARY)   (SECONDARY) *
*                                         SSID CCA LSS SSID CCA LSS*
*DEVICE   LEVEL      STATE     PATH STATUS  SERIAL#     SERIAL#    *
*------ ---------  ----------  ----------- ---------    ---------  *
* 0F61  .........  UNKNOWN...   ACTIVE..   D812 00 31   D000 00 30 *   - State UKNOWN
* SCH(0) CRIT(NO).......      CGRPLB(YES) 0000000CRB71 0000000CRB31*
*       INCRES(NO).          AUTORESYNC(YES)                      *
* PATHS PFCA SFCA STATUS: DESCRIPTION                             *
* ----- --------- ------  -------------------                     *
*   1   0133 0232   13    PATH ESTABLISHED...                     *
*       ---- ----   00    NO PATH...........                      *
*       ---- ----   00    NO PATH...........                      *
*       ---- ----   00    NO PATH...........                      *
*  SUBSYSTEM         WWNN                  LIC LEVEL              *
* -----------  ----------------          -----------              *
* PRIMARY....  5005076305FFD710          7.7.40.291               *
* SECONDARY.1  5005076305FFD70C                                   *
*******************************************************************
ANTP0001I CQUERY COMMAND COMPLETED FOR DEVICE 0F61. COMPLETION CODE: 00
```

5. In the following two volume reports for secondaries, the state is MTIR (rather than UNKNWON, as in the previous example), for Multi-Target Internal Relationship, because multi-target is enabled. Note that the order of the volume reports is different than in the previous example. The order of the volume reports is unpredictable.

```
ANTP0090I CQUERY FORMATTED LVL 6
VOLUME REPORT
************** PPRC REMOTE COPY CQUERY - VOLUME ********************
*                                          (PRIMARY)   (SECONDARY) *
*                                         SSID CCA LSS SSID CCA LSS*
*DEVICE   LEVEL      STATE     PATH STATUS  SERIAL#     SERIAL#    *
*------ ---------  ----------  ----------- ---------    ---------  *
* 0F61  PRIMARY..  MTIR......   ACTIVE..   D812 00 31   D000 00 30 *  <-- State is MTIR
* SCH(0) CRIT(NO).......      CGRPLB(YES) 0000000CRB71 0000000CRB31*
*       INCRES(NO).           AUTORESYNC(YES)                      *
* PATHS PFCA SFCA STATUS: DESCRIPTION                              *
* ----- --------- ------  ------------------                       *
*   1   0133 0232   13    PATH ESTABLISHED...                      *
*       ---- ----   00    NO PATH...........                       *
*       ---- ----   00    NO PATH...........                       *
*       ---- ----   00    NO PATH...........                       *
*  SUBSYSTEM         WWNN                  LIC LEVEL               *
* -----------  ----------------           -----------              *
* PRIMARY....  5005076305FFD710           7.7.40.291               *
* SECONDARY.1  5005076305FFD70C                                    *
********************************************************************
VOLUME REPORT
************** PPRC REMOTE COPY CQUERY - VOLUME ********************
*                                          (PRIMARY)   (SECONDARY) *
*                                         SSID CCA LSS SSID CCA LSS*
*DEVICE   LEVEL      STATE     PATH STATUS  SERIAL#     SERIAL#    *
*------ ---------  ----------  ----------- ---------    ---------  *
* 0F61  SECONDARY  DUPLEX.CAS   ACTIVE..   D400 00 30   D812 00 31 *
* SCH(0) ...............      ........... 0000000CRB41 0000000CRB71*
*       ...........           ...............                      *
* PATHS PFCA SFCA STATUS: DESCRIPTION                              *
* ----- --------- ------  ------------------                       *
*   1   0133 0232   13    PATH ESTABLISHED...                      *
*       ---- ----   00    NO PATH...........                       *
*       ---- ----   00    NO PATH...........                       *
*       ---- ----   00    NO PATH...........                       *
*  SUBSYSTEM         WWNN                  LIC LEVEL               *
* -----------  ----------------           -----------              *
* PRIMARY....  5005076305FFD70E           7.7.40.291               *
* SECONDARY.1  5005076305FFD710                                    *
********************************************************************
ANTP0001I CQUERY COMMAND COMPLETED FOR DEVICE 0F61. COMPLETION CODE: 00
```

Querying PPRC linkage

Issue the CQUERY command with the LINKINFO option to query PPRC linkage. The linkage information that is displayed includes subchannel set, world wide node name and the IDs of primary and secondary linkage adapters. You can request formatted or unformatted output, as described in “Formatted and unformatted output” on page 331.

CQUERY formatted output for linkage information

The following output shows linkage information.

```
ANTP0259I CQUERY FORMATTED LVL 1
LINKINFO REPORT
************** PPRC REMOTE COPY CQUERY - LINKINFO ******************
*                                       DEVN SCH       WWNN         *
*                                       ---- --- ----------------   *
* PRIMARY      SECONDARY          PRI: 0F50   0 5005076303FFC061   *
* ADAPTER      ADAPTER            SEC: 0F60   0 5005076303FFC05F   *
* IDS     STAT IDS                                                  *
* ------- ---- ------------------------------------------------     *
*   0100    0  0200 0202 0330 0331                                  *
*   0030    0  0200 0202 0330 0331                                  *
*   0101    0  0200 0202 0330 0331                                  *
*   0102    3                                                       *
*   0103    3                                                       *
*   0031    0  0200 0202 0330 0331                                  *
*   0032    3                                                       *
*   0033    3                                                       *
********************************************************************
```

The fields in the output are:

DEVN
Displays the device numbers of the primary and secondary storage facility images (SFIs). If you used the SWWNN keyword on the CQUERY command, the device number for the secondary SFI is not shown.

SCH Displays the subchannel sets. If you used the SWWNN keyword on the CQUERY command, the subchannel set for the secondary SFI is not shown.

WWNN
Displays the world wide node names. If you used the SWWNN keyword on the CQUERY command, the world wide node name for the secondary SFI is not shown.

PRIMARY ADAPTER IDs
Displays the IDs of the primary adapters.

STAT Displays the status of the primary adapter, which is one of the following:

0 Found primary adapter – secondary adapter connections

1 No primary adapter - secondary adapter connections

2 Primary adapter is offline

3 Primary adapter detected a link failure

4 Primary adapter topology is invalid for PPRC

5 Primary adapter detected link, switch or secondary CU errors

6 Secondary World Wide Node Name is invalid

7 Primary adapter received a timeout when checking connections

SECONDARY ADAPTER IDS
Displays a list of the IDs for the secondary adapters for each primary adapter.

Querying PPRC paths

Issue the CQUERY command with the PATHS option to query the status of all PPRC paths established for a volume's storage control.

You can request formatted or unformatted output, as described in "Formatted and unformatted output" on page 331.

In a Multi-Target Mirror environment, that is, Multi-Target is supported by the microcode and ENABLE(PPRCMT) is specified in the DEVSUP*xx* member of PARMLIB, the formatted output reflects all of the secondary control units that are defined to the primary, which can be up to 16.

Note:

1. CQUERY command output in the SYSLOG can contain a connection code at the end of the header message and at the beginning of each of the lines associated with that connection code. The system generates this code for any multiline WTO.
2. The CQUERY PATHS and VOLUME reports may show incorrect information for the secondary subsystem WWNN in the case where that secondary is located on an ESS that does not have LIC 2.3 installed. In this case the WWNN information is not reported back correctly from the primary storage control. Do not use this WWNN information to attempt to set up paths or devices on that secondary subsystem. This could lead to an inadvertent establish on the wrong device as a secondary.

Note:

The following command queries paths established for volume 0F40:

```
CQUERY DEVN(X'0F40') PATHS
```

CQUERY unformatted output with the PATHS option

The following is an example of printed output for a CQUERY PATHS UNFORMAT command for a volume on a primary storage control. The volume has at least one path established to four secondary storage controls. Refer to message ANTP0096I for details about the reported fields.

```
15.12.00 TSU00178 ANTP0096I CQUERY UNFORMATTED LVL 3       C
PATHS REPORT
000000090383,0240,0000,
000000090461,8081,5,00110000,01,001200,01,001300,01,001400,01,
00150000,01,,,,,
000000090462,8082,1,00120100,01,,,,,,,
000000090463,8083,1,00130200,01,,,,,,,
000000090464,8084,1,00140300,01,,,,,,,
ANTP0001I CQUERY COMMAND COMPLETED FOR DEVICE 0F40.
COMPLETION CODE: 00
```

CQUERY formatted output with the PATHS option

The following example of CQUERY PATHS printed output displays data for a storage control that supports eight paths. It includes FCP pathing information. Path information for paths 5-8 would not be displayed if fewer than five paths were active, or if the storage control supported only four paths.

```
ANTP0095I CQUERY FORMATTED LVL 3 989
PATHS REPORT
*************** PPRC REMOTE COPY CQUERY - PATHS ********************
* PRIMARY UNIT: SERIAL#= 000000001711 SSID= BEC0 SS= 2107 LSS= 16  *
*              FIRST        SECOND        THIRD        FOURTH      *
*            SECONDARY     SECONDARY     SECONDARY     SECONDARY   *
*SERIAL NO: 000000001691 0000000BWGV1  ............  ............  *
* SSID LSS:   CD40 10 E220 08          .......       .......       *
*    PATHS:      1             0             0             0       *
*           PFCA SFCA S*  SAID DEST S*  SAID DEST S*  SAID DEST S* *
*           --------- --  --------- --  --------- --  --------- -- *
*        1: 0031 0202 13  0031 0200 13  ---- ---- 00  ---- ---- 00 *
*        2: ---- ---- 00  ---- ---- 00  ---- ---- 00  ---- ---- 00 *
*        3: ---- ---- 00  ---- ---- 00  ---- ---- 00  ---- ---- 00 *
*        4: ---- ---- 00  ---- ---- 00  ---- ---- 00  ---- ---- 00 *
*  SUBSYSTEM         WWNN        LIC LEVEL                         *
* -----------  ----------------                                    *
* PRIMARY....  5005076303FFC061 5.2.200.22                         *
* SECONDARY.1  5005076303FFC05F                                    *
* SECONDARY.2  5005076304FFC441                                    *
*                                                                  *
* S* = PATH STATUS:                                                *
* 00=NO PATH          01=ESTABLISHED ESCON   02=INIT FAILED        *
* 03=TIME OUT         04=NO RESOURCES AT PRI 05=NO RESOURCES AT SEC*
* 06=SERIAL# MISMATCH 07=SEC SSID MISMATCH   08=ESCON LINK OFFLINE *
* 09=ESTABLISH RETRY  0A=PATH ACTIVE TO HOST 0B=PATH TO SAME CLUSTR*
* 10=CONFIG ERROR     FF=UNABLE TO DETERMINE                       *
* 13=ESTABL FIBRE PTH 14=FIBRE PATH DOWN     15=FIBRE RETRY EXCEED *
* 16=SEC ADPTR INCPBL 17=SEC ADPTR UNAVAIL   18=FIBRE LOGIN EXCEED *
* 1B=FIBRE DEGRADED   1C=FIBRE REMOVED                               *
********************************************************************
ANTP0001I CQUERY COMMAND COMPLETED FOR DEVICE 0F40. COMPLETION CODE: 00
```

The printed output includes the following path status (S*) information at the bottom of the printout:

```
S* = PATH STATUS:                                                  *
* 00=NO PATH          01=ESTABLISHED ESCON   02=INIT FAILED        *
* 03=TIME OUT         04=NO RESOURCES AT PRI 05=NO RESOURCES AT SEC*
* 06=SERIAL# MISMATCH 07=SEC SSID MISMATCH   08=ESCON LINK OFFLINE *
* 09=ESTABLISH RETRY  0A=PATH ACTIVE TO HOST 0B=PATH TO SAME CLUSTR*
* 10=CONFIG ERROR     FF=UNABLE TO DETERMINE                       *
* 13=ESTABL FIBRE PTH 14=FIBRE PATH DOWN     15=FIBRE RETRY EXCEED *
* 16=SEC ADPTR INCPBL 17=SEC ADPTR UNAVAIL   18=FIBRE LOGIN EXCEED *
* 1B=FIBRE DEGRADED   1C=FIBRE REMOVED                               *
********************************************************************
```

For a description of the fields, refer to “Fields in formatted output” on page 332.

The path status codes are defined as follows. **Path codes noted as “LSS only” are valid only for storage subsystems that support logical storage subsystems.**

00 — NO PATH
: The request to establish this path is in process, or you have not requested this entry. (The SAID/DEST appears as “----” until the CESTPATH command operation has completed. Reissue the CQUERY command with the PATHS option.)

01 — ESTABLISHED
: Path is operational.

02 — INIT FAILED
: An attempt to establish this path has failed. Verify that the physical link,

including ESCD ports, is correct for the connection you are attempting. Verify the integrity of the fiber-optic links between the requested source and the destination.

03 — TIME OUT
An attempt to establish this path has failed. Verify the physical link at the primary storage control. Determine if non-PPRC system activity has degraded link response times during the establish time. Retry the CESTPATH command.

04 — NO RESOURCES AT PRI
An attempt to establish this path has failed. Verify that the primary storage control does not already have the maximum number of logical paths already established. Verify the configuration and remove any unnecessary channel paths. When the problem is resolved, issue a CQUERY command with the PATHS option to verify that the path has been automatically established.

05 — NO RESOURCES AT SEC
An attempt to establish this path has failed. Verify that the recovery site storage control does not already have the maximum number of logical paths already established. Verify the configuration and remove any unnecessary channel paths. When the problem is resolved, issue a CQUERY command with the PATHS option to verify that the path has been automatically established.

06 — SERIAL # MISMATCH
The recovery site storage control specified by the SAID/DEST does not have the same serial number specified in the CESTPATH command. Verify the SAID/DEST port and the serial number parameters. A CQUERY command with the PATHS option from the host to a secondary volume can provide the serial number information.

07 — SEC SSID MISMATCH
(LSS only) An attempt to establish this path has failed. Verify that the path is available and is properly defined, and then retry the CESTPATH command.

08 — ESCON LINK OFFLN
(LSS only) An attempt to establish this path has failed. Retry the CESTPATH command after you have made a path available.

09 — ESTABLISH FAILED
(LSS only) An attempt to establish this path has failed. If the automatic attempts to reconnect are not successful, reissue the CESTPATH command.

0A — PATH ACTIVE TO HOST
(LSS only) An attempt to establish this path has failed. The system adapter has already established a link to the system server on this path. Retry the CESTPATH command, specifying another path.

0B — PATH TO SAME CLUSTR
(LSS only) An attempt to establish this path has failed because there is already an established path from this primary cluster to this secondary cluster.

10 — CONFIGURATION ERROR
The SAID is not valid for the primary site storage control ESCON SA card, or a PPRC path has already been established on this logical link. Verify the SAID is appropriate for the number of physical ESCON ports installed on the primary storage control SA card. Also, determine if this is already a

PPRC logical path, possibly established in the other direction. To do this, issue a CQUERY command with the PATHS option to the indicated storage controls at both sites.

13 — Fibre Path Established

14 — Fibre Channel Path Link Down

15 — Fibre Channel Path Retry Exceeded

16 — Fibre Channel Path Secondary Adapter not PPRC Capable
Some possible reasons for this status displaying are as follows:

- Secondary adapter is not configured properly, or does not have the correct version of microcode loaded.
- The secondary adapter is already a target of 32 different ESS boxes.

17 — Fibre Channel Path Secondary Adapter not Available

18 — Fibre Channel Path Primary Login Exceeded

19 — Fibre Channel Path Secondary Login Exceeded

1B — FIBRE DEGRADED
Fibre Channel Path established, but degraded due to high failure rate.

1C — FIBRE REMOVED
Fibre Channel Path removed with high failure rate.

CQUERY formatted output with the PATHS option in a Multi-Target configuration

The following example shows a portion of a CQUERY PATHS result with more than four secondary LSSes defined to the primary.

```
CQUERY FORMATTED LVL 4
PATHS REPORT
*************** PPRC REMOTE COPY CQUERY - PATHS ********************
* PRIMARY UNIT: SERIAL#= 0000000CCH81 SSID= D000 SS= 2107 LSS= 30  *
*                                                                  *
*          SECONDARY.1   SECONDARY.2   SECONDARY.3   SECONDARY.4   *
*          ------------  ------------  ------------  ------------  *
*SERIAL NO: 0000000CCH81  0000000CCH81  0000000CCH81  0000000CCH81 *
* SSID LSS:   D012 30       D012 31       D358 41       D38A 42    *
*    PATHS:      1             2             1             1       *
*          SAID DEST S*  PFCA SFCA S*  PFCA SFCA S*  PFCA SFCA S* *
*          --------- --  --------- --  --------- --  --------- -- *
*       1: 0232 0303 07  0303 0302 13  0233 0303 13  0303 0233 13 *
*       2: ---- ---- 00  0302 0303 13  ---- ---- 00  ---- ---- 00 *
*       3: ---- ---- 00  ---- ---- 00  ---- ---- 00  ---- ---- 00 *
*       4: ---- ---- 00  ---- ---- 00  ---- ---- 00  ---- ---- 00 *
*                                                                  *
*          SECONDARY.5   SECONDARY.6   SECONDARY.7   SECONDARY.8   *
*          ------------  ------------  ------------  ------------  *
*SERIAL NO: 0000000CCH81  0000000CCH81  0000000CCH81  0000000CCH81 *
* SSID LSS:   D264 3D       D2A8 3E       D2EC 3F       D024 32    *
*    PATHS:      1             1             1             1       *
*          PFCA SFCA S*  PFCA SFCA S*  PFCA SFCA S*  PFCA SFCA S* *
*          --------- --  --------- --  --------- --  --------- -- *
*       1: 0303 0232 13  0302 0233 13  0232 0233 13  0302 0303 13 *
*       2: ---- ---- 00  ---- ---- 00  ---- ---- 00  ---- ---- 00 *
*       3: ---- ---- 00  ---- ---- 00  ---- ---- 00  ---- ---- 00 *
*       4: ---- ---- 00  ---- ---- 00  ---- ---- 00  ---- ---- 00 *
*                                                                  *
*          SECONDARY.9   SECONDARY10   SECONDARY11   SECONDARY12   *
*          ------------  ------------  ------------  ------------  *
*SERIAL NO: 0000000CCH81  0000000CCH81  0000000CCH81  0000000CCH81 *
* SSID LSS:   D061 33       D09E 34       D0F4 35       D14A 36    *
*    PATHS:      1             1             1             1       *
*          PFCA SFCA S*  PFCA SFCA S*  PFCA SFCA S*  PFCA SFCA S* *
*          --------- --  --------- --  --------- --  --------- -- *
*       1: 0232 0302 13  0232 0303 13  0232 0303 13  0232 0303 13 *
*       2: ---- ---- 00  ---- ---- 00  ---- ---- 00  ---- ---- 00 *
*       3: ---- ---- 00  ---- ---- 00  ---- ---- 00  ---- ---- 00 *
*       4: ---- ---- 00  ---- ---- 00  ---- ---- 00  ---- ---- 00 *
*                                                                  *
*          SECONDARY13   SECONDARY14   SECONDARY15   SECONDARY16   *
.
.
.
*  SUBSYSTEM         WWNN                  LIC LEVEL               *
* -----------  ----------------           -----------              *
* PRIMARY....  5005076305FFD2A9           7.7.27.12                *
* SECONDARY.1  5005076305FFD2A9                                    *
* SECONDARY.2  5005076305FFD2A9                                    *
* SECONDARY.3  5005076305FFD2A9                                    *
* SECONDARY.4  5005076305FFD2A9                                    *
* SECONDARY.5  5005076305FFD2A9                                    *
* SECONDARY.6  5005076305FFD2A9                                    *
* SECONDARY.7  5005076305FFD2A9                                    *
* SECONDARY.8  5005076305FFD2A9                                    *
* SECONDARY.9  5005076305FFD2A9                                    *
* SECONDARY10  5005076305FFD2A9                                    *
* SECONDARY11  5005076305FFD2A9                                    *
* SECONDARY12  5005076305FFD2A9                                    *
* SECONDARY13  5005076305FFD2A9                                    *
* SECONDARY14  5005076305FFD2A9                                    *
* SECONDARY15  5005076305FFD2A9                                    *
* SECONDARY16  5005076305FFD2A9                                    *
.
.
.
```

Example: CLIST to add a PPRC path, add a volume pair, and query PPRC

You can process the following TSO CLIST in a standard TSO environment. This example shows how to use a CLIST to add a PPRC path, add a volume pair to PPRC, and then query the status of PPRC.

```
CONTROL PROMPT
CESTPATH                         +
   DEVN(X'0143')                 +
   PRIM(X'6060' 62019)           +
   SEC(X'6061' 68006)            +
   LINK(X'00008200')
CESTPAIR                         +
   DEVN(X'0143')                 +
   PRIM(X'6060' 62019 X'03')     +
   SEC(X'6061' 68006 X'02')      +
   MODE(COPY)                    +
   PACE(2)                       +
   CRIT(N)
CQUERY DEVN(X'0143')
```

Suspending PPRC volume pair operations

Issue the CSUSPEND command to suspend PPRC operations between the primary and secondary of a volume pair. See "CSUSPEND – suspending volume pairs" on page 316 for command specifies.

Example: The following command suspends PPRC operations between a volume pair. You must issue the CSUSPEND command to a valid primary or secondary PPRC device.

```
CSUSPEND DEVN(X'0F40') PRIM(X'6060' 62019 X'00')
              SEC(X'6061' 68006 X'09')
```

Issue a CESTPAIR command with the RESYNC option to resume PPRC operations for the volume pair.

Note: When you reestablish a suspended pair that was originally established with CRIT(YES) specified, you must again specify CRIT(YES) with the CESTPAIR command if you want this option to remain in effect.

Deleting a PPRC volume pair

Issue the CDELPAIR command to delete PPRC primary and secondary volumes. Direct the command to the primary device.

Before issuing a CDELPAIR command, verify that the necessary paths are active between the primary and secondary volume pair. If all of the paths that were established between both of the storage controls have been disabled, the CDELPAIR command will fail. PPRC may issue the following two messages:

```
ANTP0122I PRIMARY CU TIMED OUT ATTEMPTING TO COMMUNICATE
          WITH SECONDARY
ANTP0001I CDELPAIR COMMAND UNSUCCESSFUL FOR DEVICE nnnn.
          COMPLETION CODE: 08
```

After the CDELPAIR command times out, the state of the primary volume is simplex. The secondary volume remains in its previous state (duplex, pending, or suspended). You then need to issue a CRECOVER command to return the secondary volume to the simplex state.

The following command deletes a PPRC volume pair:

```
CDELPAIR DEVN(X'0F40') PRIM(X'6060' 62019 X'00')
              SEC(X'6061' 68006 X'09')
```

You can process the following TSO CLIST in a standard TSO environment.

Example: This example shows how to use a CLIST to delete a volume pair.

```
     CONTROL PROMPT
     CDELPAIR                              +
        DEVN(X'04C6')                      +
        PRIM(X'6060' 62019 X'06')          +
        SEC(X'6061' 68006 X'09')
     ENDDATA
```

If you remove a pair that is part of a multi-target configuration, the microcode also automatically removes the MTIR relationship between the secondaries.

Deleting PPRC paths

Use the CDELPATH command to delete all established paths between a primary site logical subsystem or storage control and a recovery site logical subsystem or storage control. The command only affects active paths to the recovery site logical subsystem or storage control, but not other paths to other logical subsystems or storage controls. Issue the CDELPATH command to a valid PPRC primary device, only.

Before issuing a CDELPATH command, issue a CDELPAIR command to all active PPRC volume pairs. All pair relationships must be disconnected before a path can be deleted. The CDELPATH command might cause the issuance of an ANTP0121I message if you do not follow this sequence.

The following command examples demonstrate how to delete all established paths between a primary site storage control and a recovery site storage control:

```
CDELPATH DEVN(X'0F40') PRIM(X'6060' 62019) SEC(X'6061' 68006)
CDELPATH DEVN(X'0F40') PRIM(X'6060' 5005076300CABMF0)
                                         SEC(X'6061' 5005076300CBMAC0)
```

You can process the following TSO CLIST in a standard TSO environment.

Example: This example shows how to use a CLIST to delete all PPRC paths between two storage controls.

```
CONTROL PROMPT
   CDELPATH                    +
      DEVN(X'0F40')            +
      PRIM(X'6060' 62019)      +
      SEC(X'6061' 68006)
   ENDDATA
```

For additional information about how to remove selected paths using the CESTPATH parameter, refer to "CESTPATH – establishing paths" on page 304.

Using PPRC with FlashCopy

You can combine PPRC and FlashCopy functions to generate a third copy of a set of volumes in a PPRC configuration. You can, for example, use this third copy for testing your disaster recovery configuration and creating a point-in-time consistent backup copy of a database.

The following procedure provides a method of using PPRC and FlashCopy to create a backup copy of a volume:

1. Suspend the PPRC volume pair using the CSUSPEND command.
2. Issue the CRECOVER command to recover the secondary volume and return it to SIMPLEX state.
3. Vary the secondary volume online.
4. Invoke FlashCopy through DFSMSdss to take a copy from the secondary volume.
5. Vary the secondary volume offline.
6. Issue the CESTPAIR command with the RESYNC parameter to copy any changed track to the secondary volume.

Managing PPRC extended distance mode

PPRC has been enhanced to include PPRC extended distance (XD). When the PPRC extended distance feature is installed and enabled on the ESS, it provides an alternative to synchronous PPRC. It also adds a new alternative in the ESS-to-ESS data mirroring capabilities. When PPRC extended distance is enabled, updates made to a PPRC primary volume are sent to a secondary volume asynchronously. This helps minimize the impact on application throughput.

When the PPRC extended distance feature is enabled, the primary and recovery storage control sites can be separated by very long distances. Channel extender technology can be used to link the primary and recovery storage control sites. This technology allows these systems to be connected over large distances by using less expensive telecommunication lines, with little performance impact.

Therefore, if your environment has bandwidth restrictions or has extended distance requirements, you might want to consider PPRC extended distance. When operating in extended distance mode, the primary volume sends a periodic, incremental copy of updated tracks to the secondary volume instead of a constant stream of updates. This causes less impact to application writes for primary volumes and less demand for bandwidth resources, while allowing a more flexible use of the available bandwidth. In addition, the lower bandwidth requirement creates the potential for worldwide distance capability when using channel extenders.

The examples that are provided in later sections describe how to use the PPRC extended distance feature when using TSO commands or the ANTRQST API. You can also perform PPRC extended distance-related tasks by using the IBM TotalStorage ESS Specialist Web interface or the ESS Copy Services command-line interface.

For additional information about performing PPRC extended distance-related tasks, refer to the *IBM TotalStorage Enterprise Storage Server Web Interface User's Guide.*

Understanding how PPRC extended distance works

During an extended distance operation, the ESS captures information about updates to the primary and periodically sends those updates to the secondary. After the initial copy of tracks, the ESS periodically starts a synchronization cycle where all updated tracks, in ascending order from the lowest numbered track, is copied from the primary volume to the secondary volume.

The ESS updates the secondary tracks with the current information for that track, regardless of the number of updates between the time it was last copied, the current time, and the order in which the updates occurred. When this process completes, the cycle is repeated.

There is little response time degradation on application I/O while operating in extended distance mode. Write updates to the primary volume receive an immediate completion because the synchronization cycle is independent of the primary write updates.

Setting up your environment to use PPRC extended distance

There are three requirements for setting up your environment to use PPRC extended distance mode:

- A PPRC link must be active between the primary logical subsystem or storage control and recovery logical subsystem or storage control.
- A PPRC volume pair must be established with the CESTPAIR OPTION(XD) command.
- The primary and secondary devices must be in SIMPLEX state or in SUSPEND state at the beginning of the process.

For additional information about:

- Establishing PPRC links, refer to "Ensuring PPRC paths are active."
- PPRC volume pairs, refer to "Establishing PPRC volume pairs" on page 355.
- Implementing PPRC extended distance, refer to "Scenarios using PPRC extended distance and synchronous mode for backup purposes" on page 358.

Ensuring PPRC paths are active

Before enabling PPRC extended distance, ensure that a PPRC path is active between the primary logical subsystem or storage control and the recovery logical subsystem or storage control. The establish path process, defined with the CESTPATH command, establishes a link to allow data to be transferred between the logical subsystems or storage controls.

You can determine whether a PPRC path is active by issuing the CQUERY command with the PATHS option to query the status of all PPRC paths established for a volume's storage control.

For additional information about:

- Establishing paths, refer to "CESTPATH – establishing paths" on page 304
- PPRC configuration, refer to "Establishing the PPRC solution" on page 268.
- Querying the status of all PPRC paths, refer to "Querying PPRC paths" on page 345.

Establishing PPRC volume pairs

The CESTPAIR command allows you to establish a PPRC volume pair using PPRC extended distance mode and then transition the volume pair to synchronous mode. The command includes an OPTION parameter with the following values:

- **SYNC:** This value establishes a PPRC volume pair in synchronous mode. In this mode, the volume pair reaches DUPLEX status after the initial copy of the volume pair is completed and the primary updates are mirrored synchronously with the I/O in progress. As a consequence, there is propagation overhead, manageable under 103 km, but which becomes progressively significant at very long distances. In this case, the overhead slows down the application throughput too much to meet reasonable production objectives.

  If a volume pair is established in extended distance mode, specifying this value causes the volume pair to change to synchronous mode.
- **XD:** This value establishes a PPRC volume pair in extended distance mode. In this mode, the volume pair does not reach DUPLEX state after the initial copy of the volume pair is completed. The volume remains in a PENDING.XD state. The primary volume updates complete before they are mirrored to the secondary volume. These track updates are recorded into a bitmap and periodically copied to the secondary volume. As a consequence, there is no guarantee that application dependent writes are transferred in the same sequence that they have been applied to the primary volume. Because the contents of the copy are uncertain, the secondary copy is described as being *fuzzy*, as shown in "Converting from one PPRC volume state to another" on page 356.

  The PPRC-XD solution provides tier 4 protection, which means it is a low cost solution for disaster protection, but the recovery point objective (RPO) is based on how often the tertiary FlashCopy of the secondary volumes are created. Most customers find that once a day consistency provides the protection they require with this solution. Since the application system I/Os must be quiesced or stopped until the synchronization point is achieved, PPRC-XD cannot provide protection above the tier 4 level.

  Because mirroring in extended distance mode is done asynchronously with PPRC volume pairs, CRIT(YES) is not valid for a volume pair established with OPTION(XD).

For additional information about the CESTPAIR command, refer to Chapter 14, "Peer-to-Peer Remote copy command descriptions," on page 291.

Using PPRC extended distance

The introduction of PPRC extended distance provides some of the benefits described in the following sections:

- "Using synchronous PPRC with PPRC extended distance"
- "Long-distance data migration" on page 356
- "Periodic point-in-time backups" on page 356

Using synchronous PPRC with PPRC extended distance

The enhanced flexibility with synchronous PPRC means that you can have some volume pairs established in synchronous mode and others in PPRC extended distance mode. For example, if you have an environment with restricted bandwidth but have volumes that are critical to your operation, you can establish those volumes in synchronous mode. Synchronous mode allows the recovery storage control to remain current with the primary storage control, which is critical should a disaster occur. Less critical volumes can be established in PPRC extended distance mode.

In addition, you might also consider using PPRC in synchronous and extended distance modes of operation during peak and nonpeak hours. For example, you might consider using PPRC extended distance mode during peak work hours when minimal performance impact to primary applications is an important requirement. Then, during nonpeak hours, you can use synchronous mode, which allows your data to remain current at the recovery site in the event that an unplanned outage or disaster occurs.

For additional information about when to use PPRC extended distance mode and synchronous mode, refer to "Using extended distance and synchronous modes during peak and nonpeak operations" on page 360.

Long-distance data migration

You can use PPRC extended distance for long-distance data migration. During the migration, application updates from the primary to the secondary can still occur. When the migration copy is complete you can use the new volumes to start your applications at the new location.

Periodic point-in-time backups

You can use a combination of PPRC extended distance and FlashCopy operations to create a consistent copy of the secondary volumes using the procedures described in "Scenarios using PPRC extended distance and synchronous mode for backup purposes" on page 358.

Guideline: As you plan to use PPRC extended distance for point-in-time backup solutions, ensure that for each PPRC target volume that you use in each LSS (at the secondary), there is a another volume (physically) present in each LSS in order to facilitate FlashCopy operations to target volumes.

Converting from one PPRC volume state to another

You can convert PPRC volume pairs from PPRC extended distance mode to synchronous mode and vise versa. A switch to synchronous mode can be done using the CESTPAIR command while the volume pair is active or after the volume pair has been suspended. Figure 25 on page 357 and Table 44 on page 357 explain the commands used to convert between PPRC states, including PPRC extended distance. Each number in the figure is listed in the table with the appropriate command for that state change.

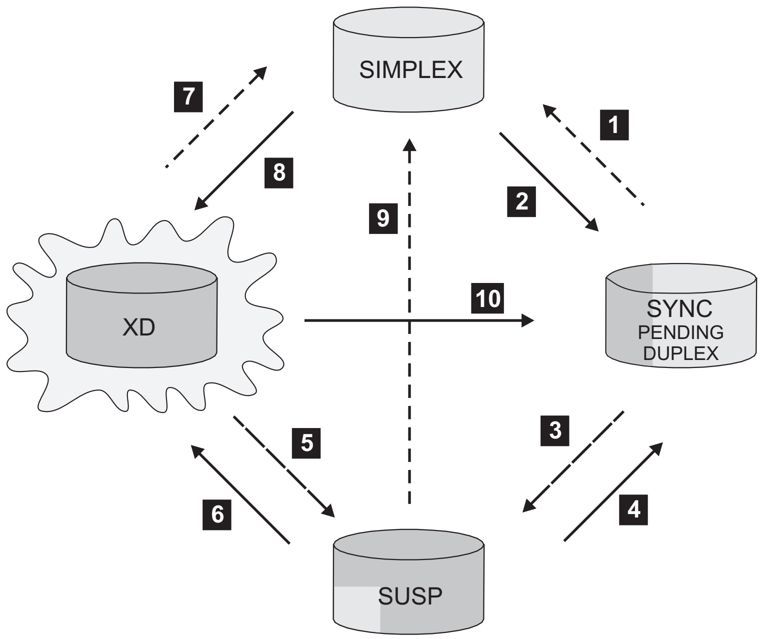


Figure 25. Volume state transitioning

Table 44. Transitioning between volume states

	To transition from...	To...	Use the following command:
1	SYNC	SIMPLEX	CDELPAIR
2	SIMPLEX	SYNC	CESTPAIR OPTION(SYNC)
3	SYNC	SUSP	CSUSPEND
4	SUSP	SYNC	CESTPAIR MODE(RESYNC)
5	XD	SUSP	CSUSPEND
6	SUSP	XD	CESTPAIR MODE(RESYNC) OPTION(XD)
7	XD	SIMPLEX	CDELPAIR
8	SIMPLEX	XD	CESTPAIR OPTION(XD)
9	SUSP	SIMPLEX	CDELPAIR
10	XD	SYNC	CESTPAIR OPTION(SYNC)

Monitoring PPRC volume pairs

To monitor the copy process of PPRC volumes, issue the CQUERY command. When PPRC extended distance is enabled, you should monitor volumes that have the most out-of-sync tracks waiting to be transmitted because of the delays introduced before updates are received by the recovery site. The amount of data lost during a disaster increases with the number of out-of-sync tracks. You can also set up automated procedures to monitor this activity. To view the output from the CQUERY command, see “Querying PPRC volumes” on page 330.

Scenarios using PPRC extended distance and synchronous mode for backup purposes

If you established a PPRC pair in extended mode and wish to convert to synchronous mode, follow the procedure in "Using extended distance operation and converting to synchronous operation." If you need to switch from synchronous mode to extended distance mode, see "Using synchronous operation and then converting to extended distance operation" on page 360.

Using extended distance operation and converting to synchronous operation

This scenario describes a procedure whereby you established the PPRC volumes pairs in extended distance mode and later want to change the pairs to a synchronous operation. The following steps outline a scenario in which you can establish pairs with the extended distance option, run in extended distance operation, synchronize the pairs, make a consistent copy of your secondary volume for backup and recovery purposes, and finally reestablish the pairs to continue operation. For more information on PPRC commands, see Chapter 14, "Peer-to-Peer Remote copy command descriptions," on page 291.

You need to be familiar with the commands that are used in these steps. For additional information about the PPRC commands, refer to Chapter 14, "Peer-to-Peer Remote copy command descriptions," on page 291.

Step 1. establish consistency at the volume level

1. Issue the CESTPAIR command with OPTION(XD) specified to establish a PPRC volume pair using the extended distance option.

```
CESTPAIR  DEVN(X'F41') PRIM(X'A763' FCA76 X'6B' X'03')
SEC(X'A762' FCA76 X'1E' X'02') MSGREQ(NO) OPTION(XD)
```

   Note:
   a. If MODE(COPY) is specified with the CESTPAIR command, the volume pair must be in SIMPLEX state. If MODE(RESYNC) is specified with the CESTPAIR command, the volume pair must be in SUSPEND or SUSP(*n*).XD state.
   b. You can issue the PPRC CESTPAIR command to an existing PPRC volume pair to cause the mode to change without losing secondary updates.
   c. Although the volume pair is established, it is not synchronized. The volume pair is in PENDING state throughout extended distance processing.
   d. While in PPRC XD mode, the secondary volumes are in a fuzzy state and cannot be used for disaster recovery.
2. Run in extended mode until a consistent copy of data is required for backup purposes, for example.
3. Issue a CESTPAIR command with OPTION(SYNC) specified to cause pending tracks and potential updates be transmitted synchronously. This process ensures that the secondary volume is consistent with the primary volume, at a point in time. The MODE parameter is ignored when you issue a CESTPAIR OPTION(SYNC) command for an active XD volume pair.

```
CESTPAIR DEVN(X'F41') PRIM(X'A763' FCA76 X'6B' X'03')
  SEC(X'A762' FCA76 X'1E' X'02') MSGREQ(NO) CRIT(YES) OPTION(SYNC)
```

4. Wait for the volume pair to reach DUPLEX state. You can monitor this activity by continuously issuing the following CQUERY command or by setting up automation to monitor when the volume pair is in DUPLEX state.

```
CQUERY DEVN(X'F41')
```

Note: At this point, you can suspend the volumes to create a point-in-time copy of all suspended volume pairs.

5. Create a data consistent copy of your secondary volume.

Step 2. freeze updates to the primary volume

Issue the CGROUP command with the FREEZE option specified (one command per storage control or LSS) to freeze I/O updates to the primary volume. Freezing your secondary volume ensures that it is consistent with the primary volume, but any updates made to the primary volume after the freeze are not reflected on the secondary volume. To freeze updates to the primary volume, issue the following command:

```
CGROUP DEVN(X'0F41') PRIM(X'A763' FCA76 X'6B' X'03') SEC(X'A762' FCA76 X'1E' X'02') FREEZE
```

Note: The CGROUP command enables the extended long-busy condition when the state of the volume pair changes to the SUSPENDED state. The application programs making updates to primary storage controls receive a long-busy condition.

Results: When you freeze I/O updates to the affected primary volume of an LSS pair, the following conditions occur:

- The I/O to primary and secondary volumes is temporarily queued
- The established paths between the LSS pairs are disabled
- The affected volume pair is suspended:
  - SYNC volume pairs in DUPLEX or PENDING states are suspended in a consistent state
  - XD volume pairs are also suspended but in a fuzzy, nonconsistent state.

Continue with "Step 3. resume operations after a freeze."

For additional information about fuzzy volume copies, refer to "Establishing PPRC volume pairs" on page 355.

Step 3. resume operations after a freeze

Following the freeze operations, issue the following CGROUP command with RUN option specified:

```
CGROUP DEVN(X'0F41') PRIM(X'A763' FCA76 X'6B' X'03') SEC(X'A762' FCA76 X'1E' X'02') RUN
```

This operation allows I/O activity to resume for the PPRC volume pair, but the pair remains suspended. However, if you do not issue the command within the extended long-busy condition timeout value, all I/O is released independently of the command that is being issued.

While the volume pair is suspended, you can copy the secondary volume for backup purposes using the procedure that is described in "Step 4. copy secondary volumes."

Step 4. copy secondary volumes

You can create a third copy of the volumes by invoking a function such as DFSMSdss to copy the data contents. Then you can copy the volumes to tape or use them for other backup and recovery operations.

At this time, secondary volumes are suspended and offline. To create a copy of the volumes, perform this procedure:

1. Issue the CRECOVER command to the secondary volumes. This step returns the secondary volumes to SIMPLEX state.

```
CRECOVER DEVN(X'0F41') PRIM(X'A763' FCA76 X'6B' X'03') SEC(X'A762' FCA76 X'1E' X'02')
```

2. Vary the secondary volumes online.
3. Use DFSMSdss to copy the secondary volumes.
4. Go to "Step 5. restart your applications."

Step 5. restart your applications

Resynchronize the PPRC primary and secondary volumes by issuing the CESTPAIR command with the RESYNC option. This process copies any changed tracks to secondary volumes that were made during the period of suspension. The following example shows the format of the command you need to issue:

```
CESTPAIR  DEVN(X'F41') PRIM(X'A763' FCA76 X'6B' X'03')
  SEC(X'A762' FCA76 X'1E' X'02') CRIT(YES)
  MODE(RESYNC) MSGREQ(NO) OPTION(SYNC)
```

If you issued the CGROUP(FREEZE) and CGROUP(RUN) commands, you must also reestablish the paths and then reestablish the volume pairs.

Using synchronous operation and then converting to extended distance operation

You can also convert your PPRC volume pairs from a synchronous copy to extended distance copy. If the PPRC volumes pairs are in a synchronous copy, you must first suspend the volume pairs. The procedure is the similar to that described in "Using extended distance operation and converting to synchronous operation" on page 358.

Using extended distance and synchronous modes during peak and nonpeak operations

You can run PPRC in a mixed mode of operation during peak and nonpeak hours. For example, you can have some volume pairs established in extended distance mode during peak working hours (08:00–18:00, for example). Then, you can convert the volume pairs to synchronous mode during nonpeak hours (18:00–8:00, for example). The following section describes the cycle of using PPRC extended distance during peak hours, converting to synchronous mode for nonpeak hours, and then converting back to PPRC extended distance during peak hours.

Establish PPRC volumes in PPRC extended distance for use during peak hours

Establish a volume pair by using the following CESTPAIR command with OPTION(XD) specified. This allows you to mirror PPRC volumes in extended distance mode during peak working hours, for example, 08:00–18:00.

```
CESTPAIR  DEVN(X'F41') PRIM(X'A763' FCA76 X'6B' X'03')
SEC(X'A762' FCA76 X'1E' X'02') MSGREQ(NO) OPTION(XD)
```

Convert PPRC volumes to synchronous mode during nonpeak hours

During nonpeak hours, you can convert the same volume pairs to synchronous mode. To do this, issue the following CESTPAIR command with OPTION(SYNC) specified. Pending tracks and potential updates will then be transmitted synchronously to secondary volumes.

```
CESTPAIR DEVN(X'F41') PRIM(X'A763' FCA76 X'6B' X'03')
  SEC(X'A762' FCA76 X'1E' X'02') MSGREQ(NO) CRIT(YES) OPTION(SYNC)
```

Convert back to PPRC extended distance mode during peak hours

If you want to later convert back to extended distance mode from synchronous mode, follow these steps:

1. Suspend the volume pairs issuing the following CSUSPEND command.

```
CSUSPEND DEVN(X'F41') PRIM(X'A763' FCA76 X'6B' X'03')
  SEC(X'A762' FCA76 X'1E' X'02') PRIMARY
```

2. Establish the volume pairs in extended distance mode by issuing a CESTPAIR command with MODE(RESYNC) OPTION(XD) specified.

```
CESTPAIR  DEVN(X'F41') PRIM(X'A763' FCA76 X'6B' X'03')
  SEC(X'A762' FCA76 X'1E' X'02')
  MODE(RESYNC) MSGREQ(NO) OPTION(XD)
```

Failover/Failback in a PPRC environment

The following scenarios describe the steps for failing over PPRC in order to move workload from the primary to the recovery site, and subsequently failing back when returning processing to the primary site. For failover/failback scenarios in a Global Mirror configuration, see “Failover/failback for a Global Mirror session” on page 443. For failover/failback scenarios in a Metro/Global Mirror configuration, see “Metro/Global Mirror recovery scenarios” on page 454.

In these discussions, the volumes at the primary (home) site are referred to as the A volumes, and the volumes at the recovery site are referred to as the B volumes. FlashCopy volumes are referred to as C volumes.

Failover process for synchronous PPRC in a planned outage

The following sequence of events would take place in a planned outage, such as might occur when preventive maintenance is performed at the primary site.

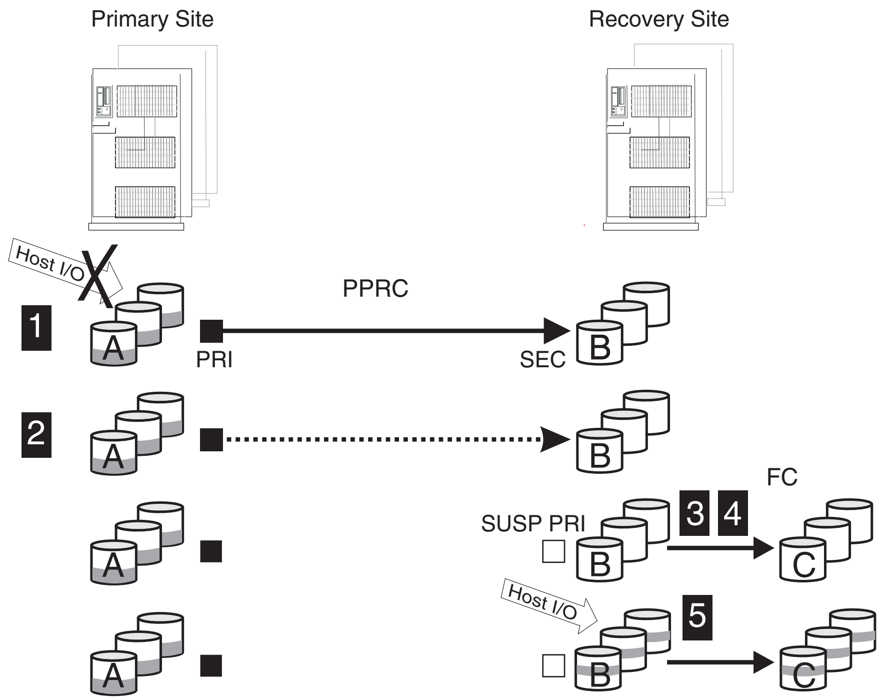


Figure 26. Failover in a Synchronous PPRC Environment

1. Quiesce applications at the primary site to reduce the risk of data loss.
2. Suspend the PPRC pairs using the PPRC Suspend command.
3. Issue the PPRC Establish Pair command to the B volume as the DEVN, specifying the FAILOVER ACTION and reversing the original PRIMARY and SECONDARY parameters. This establishes the B volumes as suspended primaries, activates change recording on the B volumes, and releases write inhibit on the B volumes.
4. If desired, FlashCopy the B volume suspended primaries to C volumes to provide safe copies for interim disaster recovery protection.
5. Restart applications at the recovery site.

Failback process for synchronous PPRC in a planned outage

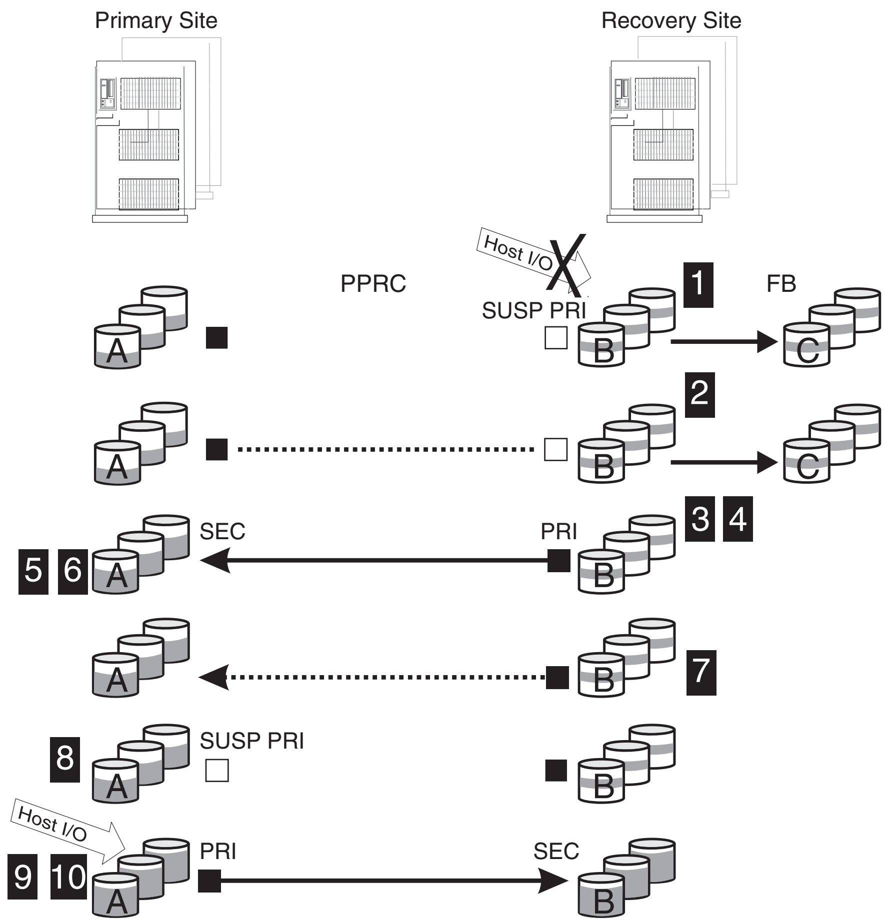


Figure 27. Failback in a Synchronous PPRC Environment

1. Quiesce applications at the recovery site to reduce the risk of data loss.
2. Use the PPRC Establish Paths command to provide a path for data flow from the recovery site to the primary site. Paths are required between all LSSs in which PPRC pairs reside.
3. Issue the PPRC Establish Pair command to the B volume as the DEVN, specifying the FAILBACK ACTION and once again reversing the original PRIMARY and SECONDARY parameters. This copies any changes recorded on the B volumes back to the A volumes
4. Withdraw the optional FlashCopy relationship at the recovery site if no longer needed.
5. Verify that paths are still established from the primary site to the recovery site. Issue PPRC Establish Paths commands if necessary.
6. Allow PPRC pairs to achieve full duplex state, indicating that all data has been copied from B to A volumes. Use the PPRC Query command to determine status or check for console message IEA494I.
7. Issue the PPRC Suspend command to the B volume as the DEVN, suspending the reversed pairs.
8. Issue the PPRC Establish Pair command to the A volume as the DEVN, specifying the FAILOVER ACTION with the original PRIMARY and

SECONDARY parameters. This reestablishes the A volumes as suspended primaries, begins change recording on the A volumes, and releases write inhibit on the A volumes.

9. Restart applications at the primary site.
10. Issue the PPRC Establish Pair command to the A volume as the DEVN, specifying the FAILBACK ACTION with the original PRIMARY and SECONDARY parameters. This resynchronizes the A and B volumes in the recovery configuration and restores the pairs to full duplex state.

Failover/Failback for synchronous PPRC an unplanned outage

The Failover process for an unplanned outage in a synchronous PPRC configuration is the same as the process for a planned outage with the single exception that step 1, quiescing the application programs at the primary site, is unnecessary.

The Failback process for an unplanned outage is identical to the process for a planned outage.

Any step in either the Failover or Failback process for an unplanned outage may require repair action beyond the scope of PPRC, in which case the assistance of a service representative might be required.

Failover process for PPRC-XD in a planned outage

In a PPRC-XD environment, data must be allowed to drain from the A volumes to the B volumes before the PPRC pair is suspended prior to initiating Failover processing.

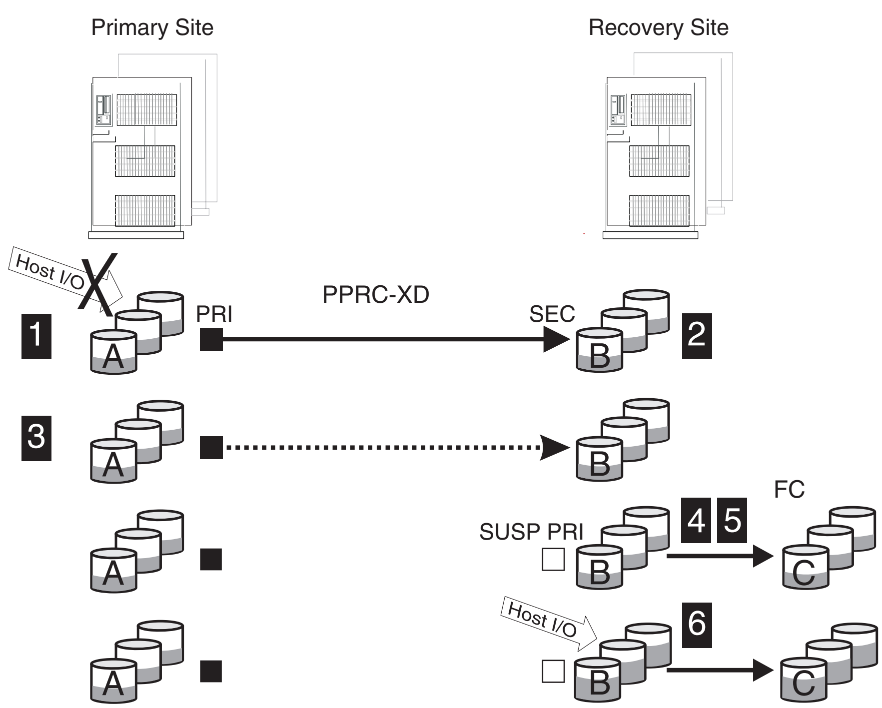


Figure 28. Failover in a PPRC-XD Environment

1. Quiesce applications at the primary site to reduce the risk of data loss.
2. Allow data to drain from A volumes to B volumes.

3. Suspend the PPRC pairs using the PPRC Suspend command.
4. Issue the PPRC Establish Pair command to the A volume as the DEVN, specifying the FAILOVER ACTION, and reversing the original PRIMARY and SECONDARY parameters. This establishes the B volumes as suspended primaries, activates change recording on the B volumes, and releases write inhibit on the B volumes.
5. If desired, FlashCopy the B volume suspended primaries to C volumes to provide safe copies for interim disaster recovery protection.
6. Restart applications at the recovery site.

Failback process for PPRC-XD in a planned outage

In a PPRC-XD environment, data must be allowed to drain from the B volumes to the A volumes before the PPRC path is established from the recovery site to the primary site.

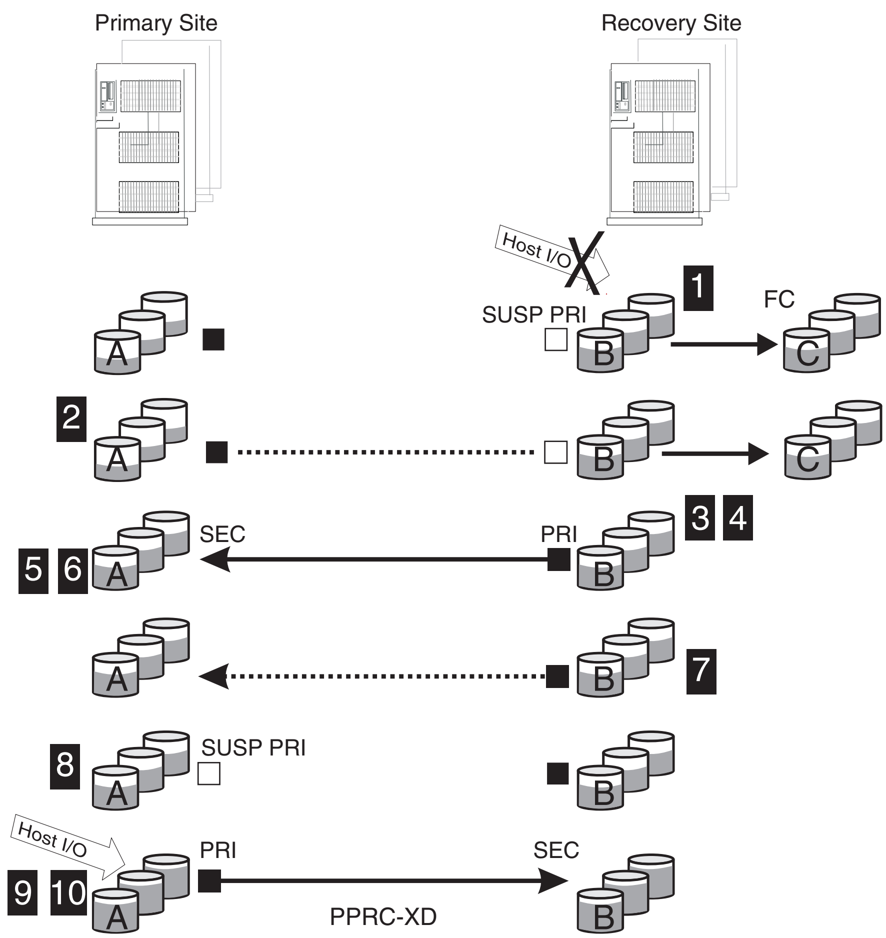


Figure 29. Failback in a PPRC-XD Environment

1. Quiesce applications at the recovery site to reduce the risk of data loss.
2. Use the PPRC Establish Paths command to provide a path for data flow from the recovery site to the primary site. Paths are required between all LSSs in which PPRC pairs reside.

3. Issue the PPRC Establish Pair command to the B volume as the DEVN, specifying the FAILBACK ACTION and reversing the original PRIMARY and SECONDARY parameters. This copies any changes recorded on the B volumes back to the A volumes
4. Withdraw the optional FlashCopy relationship at the recovery site if no longer needed.
5. Allow data to drain from B volumes to A volumes.
6. Verify that paths are still established from the primary site to the recovery site. Issue PPRC Establish Paths command if necessary.
7. Issue the PPRC Suspend command to the B volume as the DEVN, suspending the reversed pairs.
8. Issue the PPRC Establish Pair command to the A volume as the DEVN, specifying the FAILOVER ACTION with the original PRIMARY and SECONDARY parameters. This reestablishes the A volumes as suspended primaries, begins change recording on the A volumes, and releases write inhibit on the A volumes.
9. Restart applications at the primary site.
10. Issue the PPRC Establish Pair command to the A volume as the DEVN, specifying the FAILBACK ACTION with the original PRIMARY and SECONDARY parameters. This resynchronizes the A and B volumes in the recovery configuration and restores the pairs to duplex-pending-XD state.

Failover/Failback process for PPRC-XD in an unplanned outage

Because PPRC-XD does not respect the sequence of dependent writes at the secondary, the primary and secondary cannot be assumed to be in a consistent state unless the application I/O is quiesced and data is allowed to drain from the primary to the secondary. Otherwise, the secondary must be assumed to contain a “fuzzy copy”.

The only reliable method for ensuring a backup copy at the PPRC-XD recovery site is to take frequent checkpoints during which the application at the primary site is quiesced, allowing data to drain to the secondary, after which a FlashCopy is made of the secondary volumes.

Chapter 16. Peer-to-Peer Remote Copy data recovery operations

A major purpose of peer-to-peer remote copy is to enable you to switch to a recovery system in the event of a disaster to the primary system. This section presents an overview of disaster recovery from a system viewpoint and describes the procedure to direct the recovery system to take over for the primary system.

PPRC is also an excellent solution for moving data between disk devices. You might want to have a data migration procedure for planned activity, rather than for a disaster. A data migration procedure is a subset of the recovery procedure that is described below.

For additional information about using PPRC for data migration, refer to Chapter 17, “Moving and migrating data with Peer-to-Peer Remote Copy,” on page 373.

In this topic

This topic provides information about data recovery with PPRC and includes the following sections:

Examining disaster recovery from a system viewpoint

Disasters occur in many ways. In a sudden disaster, all volumes may be in normal full duplex mode when you begin recovery procedures. In such a case it is easy to determine how current the data is.

A disaster can also occur over a period of time. This rolling disaster is the most common situation, and necessitates that you check each volume to determine how current the data is. SYSLOG messages written by the ERP give the times when specific volumes are suspended.

It is important to understand how a rolling disaster can affect your primary system. This knowledge enables you to avoid a potential data integrity exposure when using PPRC in a disaster recovery situation. A disaster or series of disasters can result in intermittent failures to the primary system. In this situation, you might be unable to determine which volumes are recoverable because of one of the following scenarios:

- A disaster could leave many or all of your secondary volumes in a suspended state. Without the IEA49*xx* message to indicate when these volumes became suspended, each volume would require extensive analysis to determine how current the data is.
- If the attempt to update the secondary volume fails due to a link failure, PPRC might be unable to mark the secondary volume as suspended.

- The operating systems process all error notification actions asynchronously. Positive identification and recording of an event, such as broken PPRC ESCON links, happen after the event itself.

During a disaster recovery operation, the secondary volume would *appear* to be recoverable. There would, however, be no way to be certain that all of the data is current. The way to minimize (or even avoid) this potential data integrity exposure is by applying the disaster recovery preparations that are described in “Considering the PPRC solution” on page 267 and “Preparing for PPRC error recovery” on page 382. “Failover/failback for a Global Mirror session” on page 443 describes how to judge whether recovery data is current.

Resuming operations at the recovery site

If PPRC volumes are participating in a Global Mirror session, refer to “Failover/failback for a Global Mirror session” on page 443 for recovery procedures. Otherwise, perform the following steps to have the recovery site assume control of operations:

1. Reconfigure ESCON Directors, if necessary, to connect local systems to recovery system Disk.
2. Query each secondary disk volume to verify that it was a secondary remote copy volume and that it is in duplex state. Continue to the steps below for volume states other than duplex state.
3. If the volume is in duplex or suspended state, check the console for SYSLOG error messages that indicate that the ERP has attempted to put the volume in suspended state. If there is an error message, recover the volume as if it was in the suspended state at the time that the IEA49*xx* message indicates.
4. Issue the PPRC CRECOVER command to change the secondary volume from its current state (duplex, suspended, or pending) to simplex state in order to let that volume come online. Table 45 shows suggested peer-to-peer remote copy volume recovery procedures that are based on the state of the volume at the time of system failure.

   Rules: The PPRC recovery function must have access to each secondary volume on all recovery site storage controls at the time of the recovery.

   Some recovery system volumes may not initially be usable for recovery. Change these volume serial numbers to prevent any applications from accessing the volumes until you can either restore them, or otherwise validate their contents by recovery procedures.

Table 45. Volume failure states and recovery actions

If volume is in . . .	Then do this . . .

Duplex state	Issue the CRECOVER command, as in the following example:
	`CRECOVER DEVN(X'0F40') PRIM(X'6060' 62019 X'00')`
	`          SEC(X'6061' 68006 X'09') ID(DATA01)`
	This returns the volume to simplex state, fully recovered and ready to use. (Issue the CQUERY command again if you wish to ensure that the volume is in simplex state.) You can now vary the volume (DATA01 in the above example) online for system use. Review the steps of resuming operations at the recovery site.

Table 45. Volume failure states and recovery actions (continued)

<table>
<tr><th>If volume is in . . .</th><th>Then do this . . .</th></tr>
<tr><td>Pending state</td><td>If the volume is in pending state, then it never reached duplex state. Issue the CRECOVER command, as in the following example:<pre>CRECOVER DEVN(X'0F40') PRIM(X'6060' 62019 X'00')
         SEC(X'6061' 68006 X'09') ID(DATA01 DATA02)</pre>This returns the volume to simplex state and changes the volume serial number from its old value to a new value. (Issue the CQUERY command again if you wish to ensure that the volume is in simplex state.) You can now vary the volume (DATA02 in the above example) online to do normal recovery from backup tapes. Once the volume state is fully recovered, the volume can be changed back to its original volume serial number so that applications can allocate and use the volume.</td></tr>
<tr><td>Simplex state</td><td>Determine why the volume is in simplex state before you accept the secondary copy as current and usable. The simplex volume may either not be a remote copy volume, or the volume may be in simplex state following a previous action that ended the volume copy pair. When you are satisfied with the volume status, vary the volume online for system use.</td></tr>
<tr><td>Suspended state</td><td>Determine what caused this state to occur. If the volume was suspended by the host or primary site storage control, data on the volume is valid. How current the data is depends on the timestamp information provided in the SYSLOG IEA49xx message. If the volume was suspended by the recovery site storage control, data on the secondary volume may be downlevel or incorrect for various reasons (i.e., an equipment check). In this case, recover from the backup data volumes.

If the data on the volume is valid, issue the CRECOVER command, as in the following example:<pre>CRECOVER DEVN(X'0F40') PRIM(X'6060' 62019 X'00')
         SEC(X'6061' 68006 X'09') ID(DATA01 DATA02)</pre>This returns the volume to simplex state and changes the volume serial number from its old value to a new value. (Issue the CQUERY command again if you wish to ensure that the volume is in simplex state.)

With the volume (DATA02 in the above example) now in simplex state, vary the volume online and use the system log, database logs, and backup data volumes (when appropriate) to complete the recovery of data from the time the volume went into suspended state until the time of failure. Once the volume state is fully recovered, the volume can be changed back to its original volume serial number so that applications can allocate and use the volume.</td></tr>
</table>

5. Restart the application systems and perform the same system and application startup procedures that you perform on the primary system when you restart there following a system failure.

   Recovery system catalog entries will be consistent with those on the primary system if catalog volumes are part of the data that was copied to the recovery system. See “Copying the catalog and control data sets” on page 62 for guidelines on managing catalog updates that are not copied to the recovery system.

If the catalog change rate is very low, it may be more effective to copy the catalog with remote copy. Consider what other data sets are on the volumes with the catalogs, as PPRC will also copy these data sets to the recovery system.

PPRC failover/failback

When production is interrupted by an outage at the primary site, Failover/Failback eliminates the need to transmit all the data back from the recovery site for a full volume copy when the primary site comes back online and is ready to resume production. Following an outage, the primary volumes are often in the same state they were in when the outage occurred. In such a case, the only difference between the primary volumes and the secondary volumes are the updates made to the secondary volumes since the outage occurred.

When an outage occurs, Failover processing causes the secondary volumes to become suspended primary volumes. The Failover PPRC secondary must be in DUPLEX, SUSPENDED, or PENDING-XD state. Any subsequent changes made to those volumes are recorded by the subsystem. When the primary site is capable of resuming production, Failback processing transmits only those changes back to the primary site. The Failover process makes the following assumptions, but operational capabilities may vary and these conditions are not required for the process to function:

- All paths between the PPRC primary and secondary subsystems are inoperable
- Paths cannot be used from the secondary subsystem to the primary subsystem
- The PPRC secondary has updated data consistent with other PPRC secondary volumes

Note:

- If the device used for failover processing cannot also be used for failback (such as when the storage controller is irreparably damaged and must be replaced), then failback cannot take place and a full volume copy must be executed.
- If the secondary is in a soft fence state, failback processing clears the soft fence state.

Managing errors during recovery

This section describes errors that may occur when recovering data on the PPRC system, and the actions to take to resolve them.

Errors encountered when writing to PPRC volumes established with CRIT(YES)

An error while writing to a primary or secondary PPRC volume that was established with CRIT(YES) causes the I/O to be unit checked if there are no I/O paths available. When this occurs, there are certain error conditions where the primary is updated, but the secondary is not. You must take whatever actions are necessary to resynchronize the volumes, or recover from backup volumes.

Errors that occur in the absence of write activity

An error can occur on a primary volume that does not have any write activity at the time of the error. In this case, the volume remains in duplex state until a write occurs. To get out of this situation, send some I/O activity to the error volume, which puts it in suspended state. Then issue a CESTPAIR MODE(RESYNC) command to recover the volume.

Errors that cause a PPRC pair to become suspended

A PPRC pair may become suspended for any of the following reasons:

- Primary device write failure
- Secondary device write failure
- Secondary subsystem failure
- Communications to secondary failure
- Secondary not ready, intervention required

Either the I/O currently in process or the next I/O to the primary device of that PPRC pair is unit checked. The I/O that triggers the error may come from any attached system; therefore, the IEA49xx message can appear on any attached system's console. An operating system issues an IEA49xx console message, indicating one of the previously listed suspension reasons, based on how the system detected this failure. It does not necessarily relate to the true cause of the failure.

If you originally established the suspended PPRC pair with CRIT(YES), the system unit checks the device when the Disk ERP reissues the I/O order. You then receive an IOS message similar to the following example:

```
IOS000I 164,5D,CMD,47,0E00,00160000,VOLXXX,JOBNAME,
800000005400000F57 .....

  Sense Byte 0 '80' - command reject
  Sense Bytes 7-8 '0F57' - Device 164 in Critical State
```

An application-related error message might follow the above message.

Chapter 17. Moving and migrating data with Peer-to-Peer Remote Copy

In addition to providing data recovery capability in the event of a disaster (see Chapter 16, "Peer-to-Peer Remote Copy data recovery operations," on page 367), PPRC is an efficient tool for transferring data between various disk device types. The following are specific data transfer tasks:

- Temporarily moving data to a new location, to accommodate a repair, for instance
- Migrating data to new devices
- Making a copy of the data at any point in time

This section includes procedures for all three tasks.

When you move, migrate, or copy data with PPRC, you can put the secondary (target) disk volume on the same or on a different subsystem as the primary. Ensure that each volume is attached to a storage system with the PPRC LIC installed.

You need to be familiar with the PPRC commands before attempting to complete any of the procedures described in this chapter. For additional information about the PPRC commands, refer to Chapter 14, "Peer-to-Peer Remote copy command descriptions," on page 291.

In this topic

The following sections are included in this topic:

Moving the work load with PPRC

Perform the following steps to temporarily move data with PPRC:

1. Vary the secondary device offline to all MVS systems.
2. Issue the CESTPATH command and the CESTPAIR command with the COPY option to establish the PPRC paths and pairs.
3. Issue the CQUERY TSO command to monitor the initial copy process. (The CQUERY command returns information about the state of the PPRC volume, which changes from simplex to pending, to duplex.) During the initial copy operation, PPRC also copies all primary volume updates to the secondary (target) volume.
4. Verify that the two PPRC volumes are in the duplex state by issuing the CQUERY command.

5. Switch to the secondary (target) volume when the copy operation is complete. To switch your application from the primary volume to the secondary volume, do the following:
   a. Stop all applications that are using the primary volume.
   b. Issue the CDELPAIR command to return the PPRC primary and secondary volumes to simplex state. Optionally, you can issue a CDELPATH command if you are finished with the PPRC paths between the two storage controls.
   c. If you are moving the work loads with the intention of later copying them back, ensure that both devices have the same capacity.
   d. If the primary and secondary volumes are under the same MVS system, vary the original primary volume offline to MVS and the original secondary volume online to MVS. Both volumes have the same volume serial number, and only one can be online to MVS at a time.
   e. Establish the volumes in the reverse direction by issuing a CESTPATH command, then a CESTPAIR command with the NOCOPY option. Issue the CSUSPEND command to suspend the pair and keep a record of changed tracks.
   f. Start all applications with the new disk volumes.
   g. If you are performing this procedure to enable a device repair, make necessary repairs at this time, and verify that all data is intact on the device.
   h. When the secondary disk is again available and the data is intact, issue the CESTPAIR command with the RESYNC option to copy the changed data. If the data is suspect, establish the pair with the COPY option.
   i. Repeat these procedures when you wish to move the work load back to the original location.

Migrating data with PPRC

Perform the following steps to migrate data with PPRC:

1. Vary the secondary device offline to all MVS systems.
2. Issue the CESTPATH and CESTPAIR commands with the COPY options to establish the PPRC paths and pairs.
3. Issue the CQUERY TSO command to monitor the initial copy process. (The CQUERY command returns information about the state of the PPRC volume, which changes from simplex to pending, to duplex.) During the initial copy operation, PPRC also copies all primary volume updates to the secondary (target) volume.
4. Verify that the two PPRC volumes are in the duplex state by issuing the CQUERY command.
5. Switch to the secondary (target) volume when the copy operation is complete. To switch your application from the primary volume to the secondary volume, do the following:
   a. Stop all applications that are using the primary volume. PPRC is a synchronous copy; therefore the data on the primary volume is the same as the data on the secondary volume.
   b. Issue the CDELPAIR command to return the PPRC primary and secondary volumes to simplex state. Optionally, you can issue a CDELPATH command if you are finished with the PPRC paths between the two storage controls.
   c. If the PPRC secondary volume is a disk type containing more cylinders than the primary PPRC volume had, run ICKDSF Release 17 or above against the original PPRC secondary volume to expand the VTOC to reflect the additional space.

d. If the primary and secondary volumes are under the same MVS system, vary the original primary volume offline to MVS and the original secondary volume online. Both volumes have the same volume serial number, and only one can be online to MVS at any time.
e. Restart all applications, which now use the new disk volume.

Copying the secondary volumes with PPRC

Perform the following steps to make a copy of the secondary volumes (like volumes only) with PPRC:

1. Vary the secondary devices offline to all MVS systems.
2. Issue the CESTPATH and CESTPAIR commands with the COPY options to establish the PPRC paths and pairs.
3. Issue the CQUERY TSO command to monitor the initial copy process. (The CQUERY command returns information about the state of the PPRC volumes, which change from simplex to pending, to duplex.) During the initial copy operation, PPRC also copies all primary volume updates to the secondary (target) volumes.
4. Verify that the PPRC volumes are in the duplex state by issuing CQUERY commands.
5. When the copy operation is complete, generate a dump of the secondary (target) volumes as follows:
   a. Temporarily stop all of the applications that PPRC is copying.
   b. Issue the CSUSPEND command to the secondary volumes. PPRC now records changes to the primary volumes.
   c. Issue the CRECOVER command from the secondary host to return the secondary volumes to simplex state. Specify new volume serial numbers for the secondary volumes if the primary and secondary volumes are under the same MVS system.
   d. Vary the secondary volumes online. After the CRECOVER command completes, the secondary volumes will be in simplex state. It will be possible for applications to make updates to these volumes when they are varied online. *It is essential that no other applications write to the secondary volumes after the CRECOVER command completes.*
   e. Use DFSMSdss physical full volume dump to dump the secondary volumes.
   f. When the dumps are complete, vary the secondary volumes offline.
   g. Change the secondary volume serial numbers back to their original numbers if you changed them in 5c.
   h. Issue the CESTPAIR command with the RESYNC option to reestablish the PPRC pairs and resynchronize the volumes.

      Note: It is essential that no other applications write to the secondary volumes that DFSMSdss is dumping. If updates are made to the secondary volumes while they are in simplex state (after they are recovered), you must reestablish the pairs and resynchronize the volumes. To do this, first issue the CDELPAIR command to the primary volumes, and then issue the CESTPAIR command with the COPY option.

   If you changed the volume serial number of the secondary volume when you issued the CRECOVER command, you must restore the DFSMSdss physical full volume dump of that secondary volume. First, issue the DFSMSdss RESTORE command with the FULL and COPYVOLID keywords. Next, change the volume serial number back to the original volume serial number that the secondary volume had before you issued the CRECOVER command.

Migrating data on a secondary with Multi-Target Mirror

About this task

Starting with an existing H1 -> H2 pair, where H1 is the primary and H2 is the secondary, perform these steps to migrate the data on the secondary volume using Multi-Target Mirror.

Procedure

1. Install a new secondary volume, H2'.
2. Start Metro Mirror for the new pair, H1 -> H2'.
3. Wait for H1 -> H2' to reach full duplex. Figure 30 shows the result.

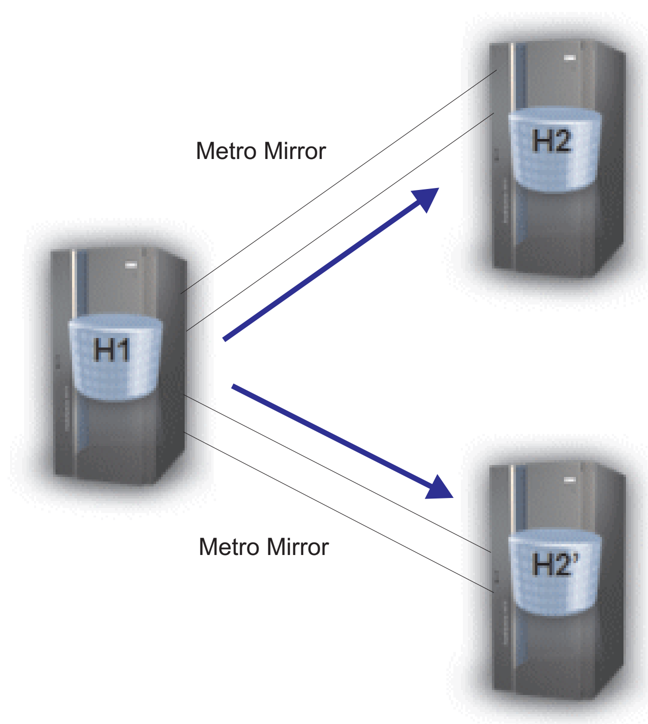


Figure 30. Configuration after Adding H2'

4. Terminate H1 -> H2.
5. Remove the original secondary volume, H2. Figure 31 on page 377 shows the result.

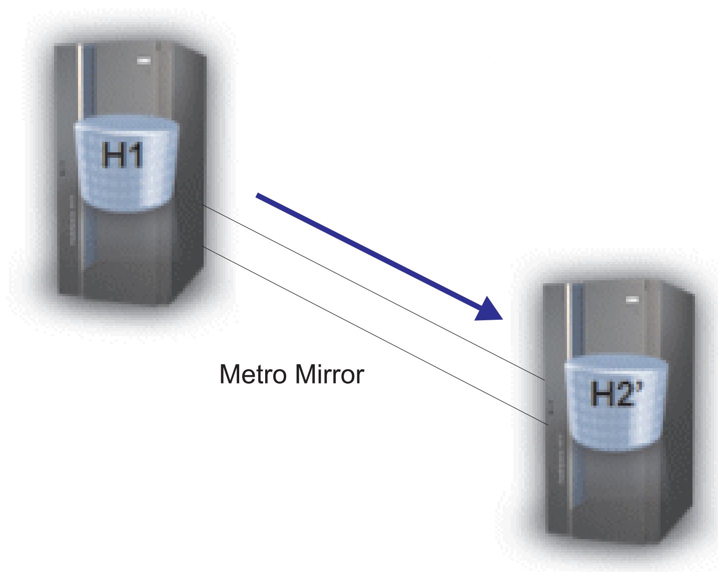


Figure 31. Configuration after Migration

Migrating data on a primary with Multi-Target Mirror

About this task

Starting with an existing H1 -> H2 pair, where H1 is the primary and H2 is the secondary, perform these steps to migrate the data on the primary volume using Multi-Target Mirror.

Procedure

1. Install a new primary volume, H1'.
2. Start Metro Mirror for the new pair, H1 -> H1'.
3. Wait for H1 -> H1' to reach full duplex. Figure 32 on page 378 shows the result.

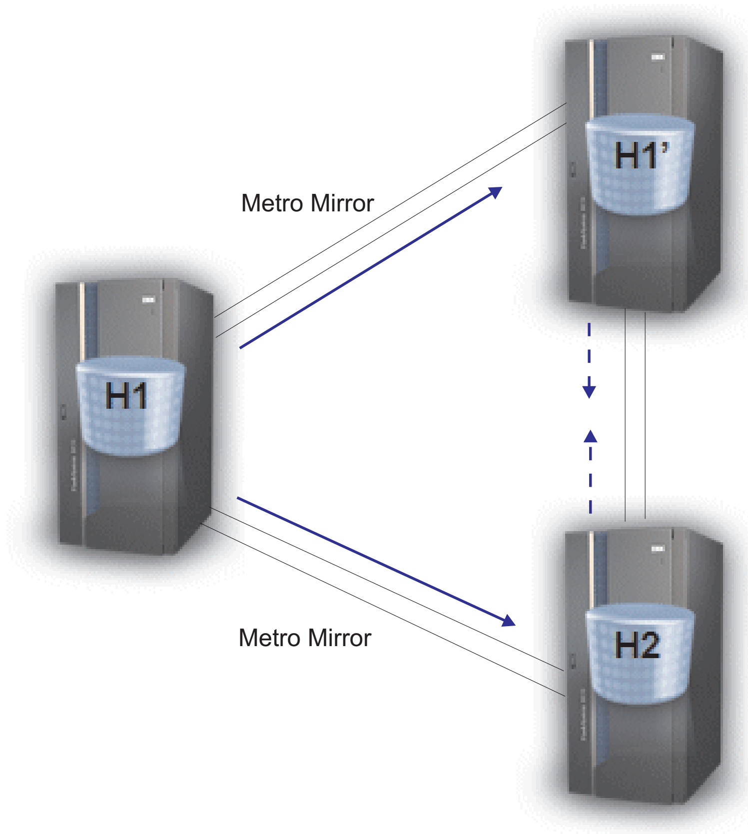


Figure 32. Configuration after Adding H1'

4. Hyperswap to H1'.
5. Resume H1' -> H2 using incremental resyncronization. Figure 33 on page 379 shows the result.

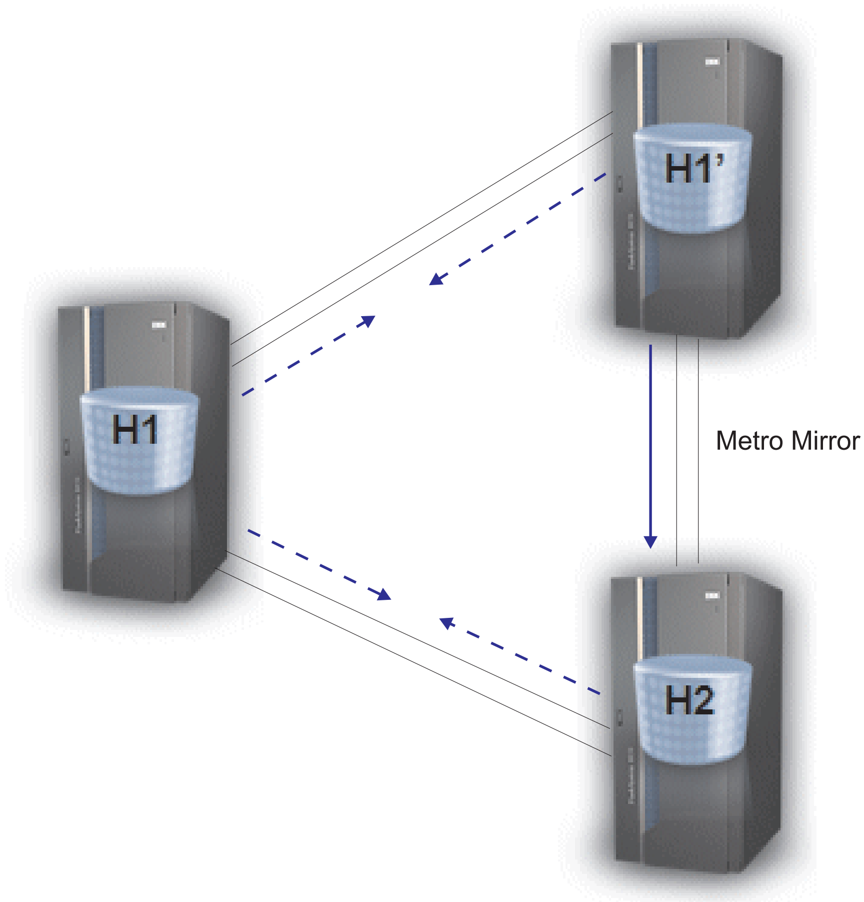


Figure 33. Configuration after Resuming H1' -> H2

6. Terminate relationships on the original primary, H1, and remove it.

Chapter 18. Recovering from Peer-to-Peer Remote Copy error conditions

This chapter describes various peer-to-peer remote copy error conditions and how to recover from them. You can usually diagnose and correct these errors independently from other PPRC session activity.

For more information about recovering with Multi-Target Mirror, including illustrations of configuration and recovery scenarios, refer to *IBM DS8870 Multiple Target Peer-to-Peer Remote Copy*.

In this topic

The following sections are included in this topic:

Understanding host error recovery procedures

There can be times when PPRC cannot replicate an update to a primary device onto the secondary device. In these cases, the primary device sends channel end, device end, and unit check status to the host that encountered the error. With the PPRC error recovery procedures (ERP) PTF that is installed, the host system performs the following actions:

1. The system stops all application I/O, *for that host*, to the primary volume in error. The host system also prevents other system images from accessing this volume.
2. The system writes an IEA49*xx* message to the log to indicate that the PPRC volume is now in suspended state or duplex state. (Ensure that the systems' log is common to both the primary system and the recovery system.)
3. The system puts information that is related to the specific failure into the SYS1.LOGREC data set for service personnel reference. SYS1.LOGREC can be common to both the primary system and the recovery system.
4. The system waits for enough time to let the IEA49*xx* message reach the recovery system. You can introduce an automation routine here that would receive control before the I/O operation can complete.
5. The system resumes all host application I/O to the primary volume. If you specified the CRIT(NO) parameter when you established the primary volume, the storage control allows subsequent I/O operations to continue without being unit checked. If you specified the CRIT(YES) parameter on the CESTPAIR command for the primary volume, then the storage control does the following:
   - Checks all subsequent write I/O operations.
   - Allows the host application to follow its own recovery actions.

   Note: An "A" in the unformatted CQUERY message ANTP0091I indicates that the volume pair is designated as CRIT(YES-ALL). This means that the pair was established with the CRIT(YES) option. The storage control maintenance panel is set to inhibit writes on any failure, which includes secondary device failures.

Preparing for PPRC error recovery

"Understanding host error recovery procedures" on page 381 outlines the actions that the host ERP starts when a PPRC duplex pair fails. It is possible that PPRC links fail such that the secondary device's storage control does not receive notices that the duplex volume is in suspended state. In this case, the primary host ERP issues an IEA49*xx* message. The host ERP action waits until the message is issued before allowing the primary application to continue. This delay gives an automation product such as NetView time to locate the IEA49*xx* message and automatically take appropriate actions.

In the event of a disaster, these actions provide data that is necessary to analyze the PPRC secondary volume to determine how current the data is. See "Putting PPRC error recovery procedures into effect" to decide which recovery actions to take. These recovery actions can include the following:

- Locate the IEA49*xx* message via NetView and copy the message to the recovery system, thereby informing the recovery system that the PPRC duplex pair is in suspended state. The message contains the timestamp that indicates the time that the pair became suspended.
- Locate the IEA49*xx* message via NetView. Next, issue a CSUSPEND command from either host to the secondary device's subsystem to notify the storage control that the PPRC pair is in suspended state. The recovery subsystem records the suspension time.

Putting PPRC error recovery procedures into effect

Table 46 identifies how the system processes an error to a volume in a volume pair depending on how the volume and path were established. The options for the CESTPATH CGROUP parameter and the CESTPAIR CRIT parameter both affect the way the recovery system processes the volume pair when a device error occurs.

Table 46. Choosing the level of volume recovery for PPRC

What level of error recovery do you want?	Specify . . .	In the event of a PPRC volume error, the ESS storage subsystem does the following . . .	Some considerations . . .
Total data consistency between primary and secondary volumes.	CRIT(YES) on CESTPAIR command (see Note).	• Suspends the volume pair that has the error. • Unit checks all write I/O to the primary device. • Issues an IEA494I message for the suspended condition.	PPRC stops all I/O to the volume pair, and suspends the pair, on any volume error.

Table 46. Choosing the level of volume recovery for PPRC (continued)

What level of error recovery do you want?	Specify . . .	In the event of a PPRC volume error, the ESS storage subsystem does the following . . .	Some considerations . . .
Automated error recovery procedures.	CRIT(YES) on CESTPAIR command, and CGROUP(YES) on CESTPATH command. **Note:** If a CGROUP FREEZE command is issued to the volume's logical subsystem when the volume is in Extended Long Busy state, the following situation occurs: • The volume status is changed to CRIT(NO). • Writes is not inhibited (unit checked) when the Extended Long Busy state ends.	• Issues an IEA494 message for the Suspending and Extended-Long-Busy state conditions. • Suspends the volume pair that has the error. • Queues application I/O, for volume pairs added with CESTPATH CGROUP(YES), in cache for two minutes. • Issues an IEA494I message for the suspended condition.	After two minutes, queued I/O resumes to the storage subsystem, if possible. Each installation is responsible for the viability of the automated recovery procedures.
A manual error recovery process.	The default for the CESTPAIR command is CRIT(NO), and the default for the CESTPATH command is CGROUP(NO).	• Suspends the volume pair that has the error. • Does **not** unit check the primary device. • Issues an IEA494I message for the suspended condition.	Each installation is responsible for the local error recovery procedures.

Note: The CRIT(YES) parameter has two modes of operation, depending on how the storage subsystem is configured. CRIT(YES-PATHS) inhibits all write operations when all paths to a secondary volume are unavailable. CRIT(YES-ALL) inhibits writes on any failure, including secondary device failures. For this reason the use of CRIT(YES) should be used only if specifically recommended by IBM support personnel.

Typically, a PPRC volume pair enters suspended state when the primary volume's storage control fails to complete a write operation to the secondary volume's storage control. The failed I/O operation receives channel end, device end, and unit check (with 'FB' indicated in sense byte 8), and the MVS host disk ERP issues an IEA49*xx* message. The NVS in the primary volume's storage control records cylinders that have changed while the volume pair is in suspended or duplex state. After you have corrected the conditions that caused the suspension, recopy these cylinders in order to return both volumes to a synchronized state.

Table 47 on page 384, Table 48 on page 384, and Table 49 on page 385 list general failures, failures detected by the host, and failures detected by the storage subsystem, and how to recover from them.

Table 47. General failure recovery procedures for PPRC

Error condition	Recovery action
Room power failure	If power is lost to the: • **Primary storage subsystem**, all volumes are placed in suspended or duplex state when power is restored. Use the CQUERY command to determine the exact state of all PPRC volume pairs. • **Secondary storage subsystem**, all volumes are placed in suspended or duplex state, and the primary volume's storage control records changes to the volume in its NVS. Issue the CESTPATH command to reestablish paths, and the CESTPAIR command with the RESYNC option to resynchronize a suspended or duplex PPRC volume pair, and to return the volumes to full duplex state. Since room power failures do not power off all devices at the same exact time, the RESYNC option is necessary in order to copy all updates that may have been made to the primary volumes, but not yet secured on the recovery site subsystem. The primary volume's subsystem maintains an NVS bitmap of all changed cylinders. This is important where power is restored to the primary hosts and primary volume's subsystem before it is restored to the secondary volume's subsystem. The CESTPAIR with the RESYNC option resynchronizes the volume pairs when power is restored to the secondary volume's subsystem.

Table 48. Host failure recovery procedures for PPRC

Error condition	Recovery action
Application host failure	1. Re-IPL the host. If power has not been lost to the subsystems, the peer-to-peer remote copy pairs will still be operational and in duplex state. 2. Issue the CQUERY command to verify the PPRC volume state for each PPRC volume pair. 3. If the volume pair is in duplex state, no further action is required. If the volume pair is in suspended or duplex state, issue the CESTPAIR command with the RESYNC option to recover the PPRC volume pair.
Last channel path failure	Host connectivity to the subsystem is lost. PPRC pair operation continues, assuming that the channel path error does not affect the PPRC link. 1. When host connectivity is restored, issue the CQUERY command to verify the PPRC volume state for each PPRC volume pair. 2. If the volume pair is in duplex state, no further action is required. If the volume pair is in suspended or duplex state, issue the CESTPATH command, and the CESTPAIR command with the RESYNC option to recover the PPRC volume pair.
Last PPRC path failure	When the last path to the recovery site storage control is lost, the PPRC pair is placed into suspended or duplex state. When the paths are repaired or reconfigured, the primary volume's storage control automatically attempts to reestablish the paths. Issue the CESTPAIR command with the RESYNC option to return the pair to duplex state.
Device varied offline	Depending on how the device is varied offline, host connectivity to the subsystem may or may not be lost. Application connectivity to the subsystem is lost, but assuming that the device has not been varied offline in "force" or "boxed" modes, the PPRC pairs continue in operation. 1. When the device is varied back online, verify the PPRC volume status with the CQUERY command. 2. If the volume pair is in duplex state, no further action is required. If the volume pair is in suspended or duplex state, issue the CESTPAIR command with the RESYNC option to recover the PPRC volume pair.

Table 48. Host failure recovery procedures for PPRC (continued)

Error condition	Recovery action
Missing interrupt	A missing interrupt generally occurs when a host issues an I/O operation to a subsystem, and the operation does not complete within a specific amount of time. 1. Analyze and correct the missing interrupt condition, using information contained in the message that the host sent to the operator console. Depending on the cause of the missing interrupt, the PPRC pair operation may or may not continue. 2. When you have corrected the missing interrupt condition, issue the CQUERY command to verify the state of the affected PPRC volume pair. 3. If the volume pair is in duplex state, no further action is required. If the volume pair is in suspended or duplex state, issue the CESTPAIR command with the RESYNC option to recover the PPRC volume pair.

Table 49. Storage subsystem failure recovery procedures for PPRC

Error condition	Recovery action
Equipment check	A number of errors fall into this category. The error may or may not be detected on the actual I/O operation that caused the failure. Generally, an equipment check that suspends a PPRC volume pair is reported as follows: • The subsequent I/O command is unit checked (with 'FB' sense) with indicators set for equipment check, permanent error, and environmental data present. • The disk ERP issues the appropriate IEA49*xx* error message. • The PPRC volume pair is placed into suspended or duplex state. For the primary PPRC volume's storage control: Change recording will *not* continue for the primary volume. When you have corrected the error condition on the primary volume's storage control, issue the CESTPAIR command with the COPY or RESYNC option to recover the PPRC volume pair. See the note at the end of this table for more information on CESTPAIR command options. For the secondary PPRC volume's storage control: Change recording *will* continue for the primary volume. When you have corrected the error condition on the secondary volume's storage control, issue the CESTPAIR command with the RESYNC option to recover the PPRC volume pair, or recopy the entire volume if its data has been corrupted by the recovery process.
Storage control failure	A number of errors fall into this category. The error may or may not be detected on the actual I/O operation that has caused the failure. Most storage control failures are reported as follows: • The I/O command that encountered the error is unit checked (with 'FB' sense) with indicators set for equipment check, permanent error, and environmental data present. • The disk ERP issues an IEA49*x*x error message. • The PPRC volume pair is placed into suspended or duplex state. For the primary PPRC volume's storage control: Change recording will*not* continue for the primary volume. When you have corrected the error condition on the primary volume's storage control, issue the CESTPAIR command with the COPY or RESYNC option to recover the PPRC volume pair. See the note at the end of this table for more information on CESTPAIR command options. For the secondary PPRC volume's storage control: Change recording *will* continue for the primary volume. When you have corrected the error condition on the secondary volume's storage control, issue the CESTPAIR command with the RESYNC option to recover the PPRC volume pair.

Table 49. Storage subsystem failure recovery procedures for PPRC (continued)

Error condition	Recovery action
Invalid track format	An invalid track format error is reported as follows: • Returned sense indicates an equipment check, permanent error, with environmental data present. • The disk ERP issues a "Volume Suspended" error message (the same action that occurs with dual copy). For the primary PPRC volume's storage control: Change recording will *not* continue for the primary volume. When you have corrected the error condition on the primary volume's storage control, issue the CESTPAIR command with the COPY or RESYNC option to recover the PPRC volume pair. See the note at the end of this table for more information on CESTPAIR command options. For the secondary PPRC volume's storage control: Change recording *will* continue for the primary volume. When you have corrected the error condition on the secondary volume's storage control, issue the CESTPAIR command with the RESYNC option to recover the PPRC volume pair.
Permanent data check	For a permanent data check on a primary PPRC volume, the PPRC volume pair status is not changed as a result of this specific error. The following actions occur: • A message is sent to the operator console. • The application program receives channel end, device end, and unit check in response to its I/O operation. • The disk ERP takes the appropriate recovery actions, including issuing the operator message and logging the error. • The application program performs its recovery actions. During a destage operation, a permanent data check on a track in the primary volume of a PPRC volume pair causes the primary volume's storage control to pin that track in NVS. A subsequent I/O command is unit checked with 'FB' sense. The disk ERP issues the appropriate IEA49*xx* error message, and the PPRC volume pair is placed into suspended or duplex state. Change recording continues for the primary volume. When the pinned data condition is corrected on the primary volume's storage control, issue the CESTPAIR command with the RESYNC option to recover the PPRC volume pair. A permanent data check on a track in a secondary volume of a PPRC volume pair causes the secondary volume's storage control to pin that track in NVS. A subsequent I/O command is unit checked with 'FB' sense. The disk ERP issues the appropriate IEA49*xx* error message, and the PPRC volume pair is placed into suspended or duplex state. Change recording continues for the secondary volume. When the pinned data condition is corrected on the secondary volume's storage control, issue the CESTPAIR command with the RESYNC option to recover the PPRC volume pair.

Table 49. Storage subsystem failure recovery procedures for PPRC (continued)

Error condition	Recovery action
Intervention required	For the primary PPRC volume, this normal system operating condition is handled the same as it is for simplex devices. The host issues an "intervention required" message for the device, then takes the appropriate actions based on the operator's reply. No data is changed on the primary volume, and the PPRC volume pair state remains unchanged. When the intervention condition is cleared, normal operations resume. For the secondary PPRC volume: • The subsequent I/O command is unit checked (with 'FB' sense) with equipment check, permanent error, and environmental data present indicators set. • The disk ERP issues an IEA49*xx* error message indicating "Intervention Required" for the secondary device. • The PPRC volume pair is placed into suspended or duplex state. Change recording *will* continue for the primary volume. When you have corrected the error condition on the secondary volume, issue the CESTPAIR command with the RESYNC option to recover the PPRC volume pair.
Cache or NVS failure, or cache reinitialization	PPRC implements write I/O operations to both primary and secondary volumes. The cache and NVS in each respective storage control hold the write updates until the updates are destaged. An NVS or cache error, or the reinitialization of cache, puts the PPRC pair into suspended or duplex state. Change recording continues. For a cache or NVS failure or cache reinitialization that affects the storage control at either end, issue the CESTPAIR command with the RESYNC option to the suspended volume after the cache and NVS have been made available. This action reestablishes the volume pair and copies any cylinders that were modified while the pair was suspended. The storage control automatically copies all cylinders from the primary volume to the secondary volume whenever an NVS failure affects the primary volume's storage control. This is necessary so that the primary volume's storage control can maintain the changed-cylinder map within its NVS.

Note: Generally, repairs to volumes are done using ICKDSF. The volume pair must be reinitialized if corrective actions change the data on a track so that the primary and secondary volume tracks are no longer identical. That track can be restored from a backup volume if the error causes data on a specific track to be overwritten. Select the appropriate CESTPAIR copy mode based on the completed repair action and the status of the data on the primary and secondary volumes.

Chapter 19. Peer-to-Peer Remote Copy dynamic address switching (P/DAS)

PPRC dynamic address switching (P/DAS) is a software function that provides the ability to redirect all application I/O from one PPRC volume to another PPRC volume with minimal application impact. P/DAS works with **PPRC volumes only**, and cannot redirect activity among XRC volumes.

P/DAS allows application-transparent switching of I/O to support the following tasks:

- Planned outages (device or subsystem)
- Device migration
- Workload movement

You can use P/DAS in shared-disk environments, including both sysplex environments and nonsysplex environments. P/DAS switching includes PAV Alias UNBIND for the original PAV base device and Alias BIND to the alternate PAV base device.

The HiperSwap function of GDPS builds on P/DAS technology but provides the total systems solution to allow application programs to continue to run in a seamless fashion following a failure of one or more primary subsystems. HiperSwap takes the P/DAS technology and creates a true business continuance solution for the customer. Visit http://www.ibm.com/servers/eserver/zseries/gdps.html for more information.

In this topic

The following sections are included in this topic:

How does P/DAS work?

P/DAS commands allow the system operator to redirect application I/Os that are currently sent to the PPRC primary volume to instead go to the secondary volume. PPRC volume pairs must be in duplex state for the P/DAS function to perform.

P/DAS manages all I/O redirection at the I/O supervisor level. P/DAS operations are therefore nearly transparent to any application program that reads from or writes to the volume. Based on the options you select, the redirection of the I/O can continue across system IPLs. The options can apply to all systems in the sysplex that are attached to the volumes.

The P/DAS function performs device switching and copying functions, as does dual copy. If you are familiar with dual copy operation, a comparison of both functions might be helpful. Table 50 summarizes some of the operating differences between the dual copy and P/DAS functions.

Table 50. Comparing P/DAS and dual copy

Function	P/DAS	Dual copy
How is switching accomplished?	Nontransparently	Transparently
Where does the operation occur?	Between storage controls	Within the same storage control
How is the function controlled?	By the operator	By the subsystem
Which system component manages the swap?	MVS system	Storage subsystem
How is I/O addressed?	Operator specifies target address	Operator specifies source address

P/DAS operates similarly to DDR Swap. After a swap operation, the "To" device comes online and the "From" device goes offline.

Preparing for P/DAS operations

This section describes the hardware and software requirements, environmental conditions, and installation-specific considerations that must be satisfied before initiating P/DAS functions.

Hardware and software requirements

The P/DAS function is available on all supported z/OS releases. P/DAS operations rely on the PPRC functions of the storage control.

Environmental conditions

The following conditions must be met before a P/DAS operation is initiated:

- The source volume for the P/DAS operation must meet the following conditions:
  - **Must be the primary volume of an active PPRC pair**.
  - Must be online to the system.
  - **Must not be part of an active XRC session**.
  - Must not have any active paging data sets in use.
  - Must not have outstanding reserves (cancel any jobs that have outstanding reserves prior to issuing P/DAS commands).
- The secondary volume for the P/DAS operation must meet the following conditions:
  - Must also be the secondary volume of the PPRC pair.
  - Must not have any outstanding target device allocations to it, such as ICKDSF or application programs.
- The PPRC pair must meet the following conditions:
  - The pair must be in duplex mode, which ensures that both the source and target devices are identical in data content.
  - The pair must be of the same device type, but not necessarily the same model.

- Both volumes must either have the same capacity, or the capacity of the target volume must be greater than that of the source volume. (See Note below.)
- The pair must not be in a CGROUP FREEZE state when you initiate P/DAS.

Note: Following the P/DAS switch operation, the additional capacity of the target volume will not be realized until you run a subsequent ICKDSF job against the target volume to rebuild the VTOC.

- Neither the SVC dump nor the ICKDSF processes can be active during P/DAS operations.

Additional P/DAS considerations

Depending on your installation and operations, consider the following items as you plan for P/DAS operations:

- Ensure that a JES3 restart or reinitialization uses the new device addresses that result from a P/DAS operation. The JES3 INISH deck must reflect the new device addresses for devices involved in a P/DAS operation. This includes devices that are part of a JES3 spool volume.
- Ensure that automatic procedures do not involve a device that P/DAS is switching, as I/O cannot occur during a switch operation.
- You can swap the SYSRES volume, provided that the volume meets the other P/DAS requirements.
- Do not use either the PPRC CSUSPEND QUIESCE option or the CGROUP FREEZE option in conjunction with P/DAS functions.
- Consider assigning a device the same device address on all systems in a sysplex environment. Operations such as P/DAS are easier when affected devices have the same device addresses within all systems.
- P/DAS supports operations between 3390, RAMAC, and ESS devices. However, you might want to run the latest level of ICKDSF to ensure that the VTOC reflects the proper device characteristics.
- The system operator must be aware of when a P/DAS function has occurred before a system IPL, and ensure that the proper volume is placed online after the IPL.

  During IPL, the system operator is prompted to select which volume to place online in situations where both the primary and secondary volumes are operational and have the same volume serial number.
- Be aware that when P/DAS reestablishes a pair in the opposite direction, it establishes the new pair with the CRIT(NO) option.

  P/DAS operates with volumes that were established with either the CRIT(YES) or CRIT(NO) keywords. P/DAS must reestablish the pair with the CRIT(NO) option because the volume pair will subsequently become suspended. To reestablish the pair with the CRIT(YES) option, issue a CSUSPEND command, then a CESTPAIR command with RESYNC and CRIT(YES) parameters.
- Do not configure P/DAS to swap between installation-static and dynamic UCBs, as the results are unpredictable.
- Do not involve volumes that contain sysplex control data sets in P/DAS volume pair operations.
- If volume pairs are in PPRC extended distance mode, convert the pairs to synchronous mode to allow them to reach DUPLEX status before a swap operation begins. This ensures that the volume pairs are in sync and all updates are copied to the secondary.

P/DAS operations

The following section describes the operator commands and available options to control P/DAS functions. P/DAS operations work within sysplex, nonsysplex, and shared-disk environments. First, we examine what a PPRC configuration looks like before P/DAS functions make changes to it.

Typical PPRC configuration before P/DAS operations

Figure 34 shows a typical PPRC configuration as it exists before issuing any P/DAS commands. You can establish PPRC paths in both directions. You do not need to use an ESCON Director. With ESCON Directors, however, PPRC pairs can be up to 43 km apart.

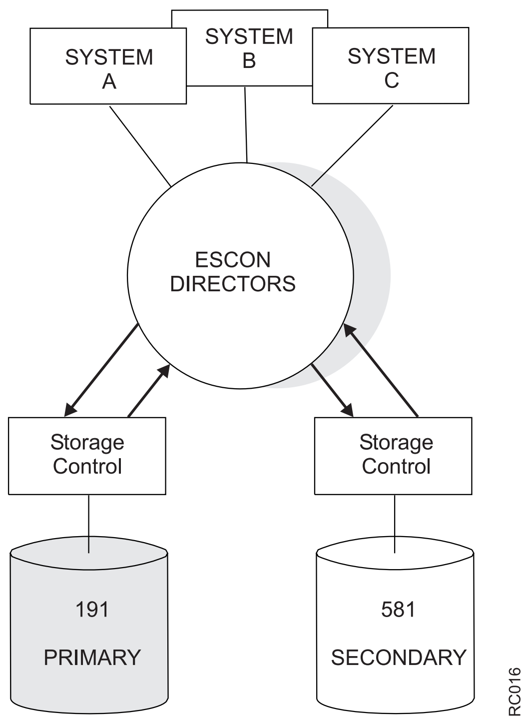


Figure 34. A typical PPRC configuration before a P/DAS operation. Disk device 191 is the PPRC primary volume, receiving updates from three attached systems, A, B and C, through a network of ESCON Directors. PPRC copies all updates to secondary volume 581.

Steps for using P/DAS in a nonsysplex, shared-disk environment

This section describes the sequence of commands for performing a P/DAS operation in a nonsysplex, shared-disk environment. In all cases, device *ssss* refers to the primary source volume. Device *tttt* refers to the secondary target volume.

Before you begin:

- Identify all systems that have access to the devices that you want to switch. *You must issue the commands that follow on all systems that are connected to the devices that you plan to switch*.
- If volume pairs are in PPRC extended distance mode, convert the pairs to synchronous mode to allow them to reach DUPLEX status before a swap operation begins. This ensures that the volume pairs are in sync and all updates are copied to the secondary.

Perform the following steps to use P/DAS in a nonsysplex, shared-disk environment:

1. Issue **IOACTION STOP,DEV=***ssss*.

   Initiate a P/DAS operation by issuing an IOACTION STOP command on *all* systems that are attached to the primary volume that is specified by DEV=*ssss*. This command requests that P/DAS stop I/O that is directed to device number *ssss*. DEV=*ssss* specifies a primary volume of a duplex PPRC volume pair.

   All application I/O that is issued to the primary volume remains queued in the MVS system until you issue an IOACTION RESUME command. P/DAS ends, with a simulated I/O error, any active I/O that does not complete within five seconds after the MVS system has accepted the IOACTION STOP. This action forces the disk error recovery procedures to retry the I/O operation.

   The MVS system issues the following messages to the operator console (or consoles) of all systems when the I/O for the primary device has been stopped on that system:

   ```
   IOS600I IOACTION - THE FOLLOWING DEVICE(S) HAVE BEEN STOPPED:
                   dev dev1-dev2...

     IOS601I IOACTION - DEVICES REMAIN IN THE STOPPED STATE.
           USE THE 'D IOS,STOP' COMMAND TO DISPLAY THE DEVICES
   ```

   When all attached systems have received the above messages, issue the swap command to switch volumes. MVS issues an IOS610I message in response to your D IOS,STOP command.

2. Issue **SWAP** *ssss*,*tttt*.

   Issue the SWAP command for the volume pairs involved in the P/DAS operation. The SWAP command directs the system to switch the source device *ssss* with the target device *tttt* and prepares the system to redirect all I/Os issued to device *ssss* to the target device *tttt*. Issue this command on all systems that are attached to the volumes that you plan to switch.

   Note: Be aware that swapping between devices that are not a PPRC volume pair is currently allowed. However, because these devices are not a PPRC volume pair, there is no guarantee that the data on both volumes match. If the swap is allowed to continue based on the reply to message (IGF523A) by the operator and the volumes do not match, the data that is yet to be copied will be lost.

   If I/O cannot be issued to the primary device of a volume pair being swapped, then the swap will not proceed even if the operator indicates for the swap to continue. I/O access to the primary and secondary volumes of the swapping pair is required.

3. P/DAS performs validation.

   Before the SWAP operation is complete, P/DAS performs validation to ensure that it can complete the swap operation. If conditions exist that can cause a data integrity exposure, P/DAS ends the operation and generates an error message. See "P/DAS error conditions" on page 400 for error message descriptions.

   P/DAS does not proceed when it cannot determine the status of both volumes, or when the volumes are not in duplex state. Ensure that the environment meets the conditions necessary to support P/DAS. "Preparing for P/DAS operations" on page 390 describes the conditions.

If the validation completes successfully, P/DAS issues the following message to prompt the operator as to how to proceed:

```
IGF520A VERIFICATION COMPLETE: REPLY 1 TERMINATE PAIR, AND SWAP
    | 2 SWITCH PAIR, AND SWAP | 3 CONTINUE SWAP | 4 TERMINATE SWAP
```

4. Reply to the **IGF520A** message.

   "Replying to P/DAS-related messages" on page 397 describes message reply options. The system takes action according to two things:
   - The option that is selected by the operator
   - The environment that exists at the time the command is issued.

   The system issues the following completion message after the swap function is complete:

   ```
   IGF505I SWAP FROM ssss TO tttt COMPLETE
   ```

   Variable *ssss* is the source device number, and *tttt* is the target device number.
5. Issue **IOACTION RESUME,DEV=***tttt*.

   In a shared-disk environment, *ensure that all attached systems have received the IGF505I message before you issue the IOACTION RESUME command*. After you have received the IGF505I message, issue an IOACTION RESUME command from all systems that are attached to device *tttt*. You must issue the command to device number *tttt* because device *tttt* is now the primary volume. The IOACTION RESUME command resumes all application I/O to device *tttt*. All application I/O that used to go to the source device *ssss* now goes to the target device *tttt* instead. This includes all I/O that is queued by MVS since the STOP command and all subsequent I/O. You must issue this command on all systems that share this volume.

   The following message received at the console indicates that all application I/O has resumed and that the P/DAS operation has completed:

   ```
   IOS607I IOACTION - THE FOLLOWING DEVICE(S) HAVE BEEN RESUMED:
                      dev,dev1-dev2
   ```

Typical PPRC configuration after a P/DAS function has completed

Figure 35 on page 395 shows what the configuration looks like after the P/DAS function has completed. I/O has been redirected to the volume that was originally the secondary of the PPRC pair. Application I/O continues without interruption. In the example shown in Figure 35 on page 395, the operator has chosen to keep both of the PPRC pair devices in operation. You can reverse PPRC operations if you have established paths in both directions.

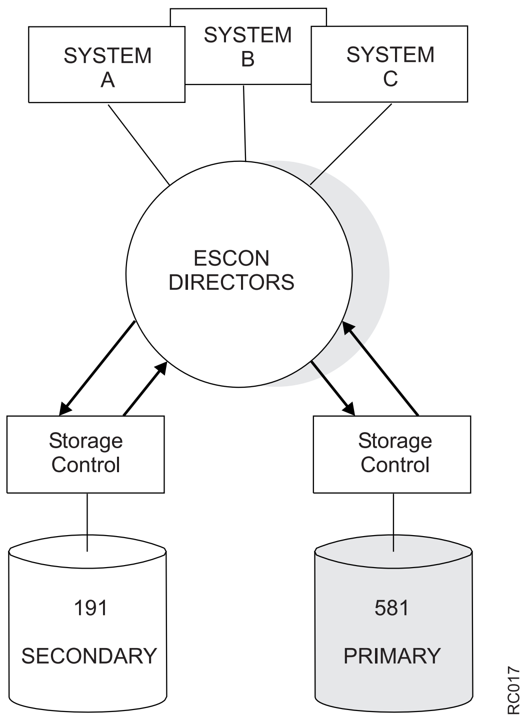


Figure 35. P/DAS swap operation complete. Disk device 581 is now the PPRC primary volume, receiving updates from attached systems through a network of ESCON Directors. PPRC copies all updates to device 191, which has now become the secondary volume.

Steps for using P/DAS in a sysplex environment

In the sysplex environment, P/DAS operations are similar to those of the nonsysplex, shared-disk environment. See "P/DAS operations example" on page 402 for an example of a sysplex scenario.

Before you begin:

- You must choose one system to be the "main system." When multiple systems are involved, you must issue the following commands on each system in the specified order.
- If volume pairs are in PPRC extended distance mode, convert the pairs to synchronous mode to allow them to reach DUPLEX status before a swap operation begins. This ensures that the volume pairs are in sync and all updates are copied to the secondary.

Perform the following steps to use P/DAS in a sysplex environment:

1. Issue **ROUTE *ALL,IOACTION STOP,DEV=***ssss*.

   Initiate a P/DAS operation by issuing this command on the "main system." This command requests that P/DAS stop I/O that is directed to device number *ssss*. ROUTE *ALL directs this action to all systems in the sysplex. DEV=*ssss* specifies a primary volume of a duplex PPRC volume pair. The MVS system stores all application I/O that is issued to the primary volume until you issue a ROUTE *ALL,IOACTION RESUME command. P/DAS ends, with a simulated I/O error, any active I/O that does not complete within five seconds after MVS has accepted the IOACTION STOP. This action forces the disk error recovery procedures to retry the I/O operation.

The MVS system issues the following messages to the operator console for all systems in the sysplex when the I/O for the primary device has been stopped on that system:

```
IOS600I IOACTION - THE FOLLOWING DEVICE(S) HAVE BEEN STOPPED:
                   dev dev1-dev2...

   IOS601I IOACTION - DEVICES REMAIN IN THE STOPPED STATE.
           USE THE 'D IOS,STOP' COMMAND TO DISPLAY THE DEVICES
```

When all attached systems have received these messages, continue with the SWAP command. MVS issues an IOS610I message in response to your D IOS,STOP command.

2. Issue **ROUTE *ALL,SWAP** *ssss*,*tttt*.

   Issue the ROUTE *ALL,SWAP command for the volume pairs involved in the P/DAS operation. ROUTE *ALL automatically directs this action to all systems in the sysplex. This command directs the systems to switch the source device *ssss* with the target device *tttt*, and prepares the systems to redirect all I/Os issued to device *ssss* to the target device *tttt*.

3. P/DAS performs validation.

   Before the SWAP operation is complete, P/DAS performs validation to ensure that it can complete the swap operation. If conditions exist that can cause a data integrity exposure, P/DAS ends the operation and generates an error message. See “P/DAS error conditions” on page 400 for error message descriptions.

   P/DAS does not proceed when it cannot determine the status of both volumes, or when the volumes are not in duplex state. Ensure that the environment meets the conditions that are necessary to support P/DAS. “Preparing for P/DAS operations” on page 390 describes the conditions.

   If the validation completes successfully, P/DAS issues the following message to prompt the operator as to how to proceed:

```
IGF520A VERIFICATION COMPLETE: REPLY 1 TERMINATE PAIR, AND SWAP
    | 2 SWITCH PAIR, AND SWAP | 3 CONTINUE SWAP | 4 TERMINATE SWAP
```

4. Select the P/DAS action.

   Attention: Do not reply to any of the system messages until all of the systems in the sysplex have issued IGF520A messages.

   Select one of the following replies from the main system console:

   - Reply with “1” to end the pair and redirect I/O to the secondary volume.
   - Reply with “2” to swap the primary and secondary volumes and redirect all I/O to the new primary volume.

   Attention: If more than one system issues the reply from the main system console, the P/DAS swap operation may fail on one or more systems. If such an error does occur, reestablish the PPRC pair on all systems and retry the P/DAS operation.

   If you choose Reply 2 (switch pair, and swap), P/DAS directs the system to take action according to the PPRC environment that is in effect at the time the command is issued, as described in Table 52 on page 399.

The main system issues the following completion message when the swap function is complete:

```
IGF505I SWAP FROM ssss TO tttt COMPLETE
```

Variable *ssss* is the source device number, and *tttt* is the target device number.

5. Reply from other sysplex systems.

   Ensure that the main system has issued an IGF505I message before replying to outstanding IGF520A messages on other sysplex systems. At this point, reply to outstanding IGF520A messages on all other sysplex systems by entering “3” (Continue Swap). Table 51 describes the resulting system action.

6. Issue **ROUTE *ALL,IOACTION RESUME,DEV=***tttt*.

   Each of the attached sysplex systems issues an IGF505I message in response to your “Continue Swap” reply. Ensure that all systems in the sysplex have received the IGF505I message, and then issue the IOACTION RESUME command to device number *tttt*. You must issue the command to device number *tttt* because device *tttt* is now the primary volume. The IOACTION RESUME command resumes all application I/O to device *tttt*. All application I/O that used to go to the source device *ssss* now goes to the target device *tttt* instead.

   This command is equivalent to the IOACTION RESUME command in the nonsysplex, shared-disk environment. The difference is that the specification of ROUTE *ALL directs this action to all systems in the sysplex. Every system in the sysplex should acknowledge this command with an IOS607I message.

   Note: P/DAS does not actually redirect and resume the application I/O to the target volume until this step has completed. Each attached system redirects the I/O as it receives and processes the IOACTION RESUME command.

Replying to P/DAS-related messages

The operator’s reply to the IGF52xA messages directs the P/DAS action, as described in Table 51. In some cases, the activities performed by P/DAS are based on a combination of the operator reply and the environment at the time that the P/DAS swap request is made.

Table 51. Reply options for IGF52xA messages

Your reply (to message) . . .	Results in this system action . . .	Notes . . .
Terminate Pair, and Swap (IGF520A)	This reply directs the system to perform the following functions: 1. End the PPRC pair and stop copy operations. Other PPRC operations for other pairs continue unchanged. 2. Redirect all application I/O from the source device *ssss* to the target device *tttt*. Application I/O is not affected by the device switch. The I/Os only update the secondary volume from this point on, and the primary and secondary volumes will no longer contain the same information.	In a **sysplex**, issue this reply from the main system only. **Application:** For device migration scenarios where the source volume is no longer going to be used (such as when the device lease will expire) and you want to change from one set of primary volumes to a second set of primary volumes.

Table 51. Reply options for IGF52xA messages (continued)

Your reply (to message) . . .	Results in this system action . . .	Notes . . .
Switch Pair, and Swap (IGF520A)	This reply redirects application I/O to the secondary volume. The system takes action according to the PPRC environment that is in effect at the time of the swap request. The action performed depends on the following two conditions: • How the PPRC paths are currently established • Whether the target device is the same size as the source device The resulting system actions are summarized in Table 52 on page 399.	In a **sysplex**, issue this reply from the main system only. **Application:** For device or subsystem maintenance, and workload movement.
Continue Swap (IGF520A, IGF521A, or IGF522A)	This reply prompts the system to redirect application I/Os from the primary volume to the secondary volume. Thus, all I/Os that were directed to the source device *ssss* are now directed to the target device *tttt*. After completion of this option, device *ssss* is no longer involved in any operation. Like dual copy, the additional space on the target device *tttt* cannot be used until an ICKDSF operation has been run against the target device to rebuild the VTOC so the additional volume capacity is recognized.	This reply is valid within sysplex and nonsysplex environments. In a **sysplex**, issue this reply from any system. **Application:** To remove a primary volume from the configuration.
Try Again (IGF521A or IGF522A)	This reply directs the system to perform the following functions: 1. End the pair, and return both devices to simplex state. It is the equivalent of issuing a CDELPAIR command for the PPRC pair. 2. Establish a new PPRC pair in the reverse direction. The new path is from the new primary (original secondary volume, device number *tttt*) to the new secondary (original source volume, device number *ssss*). This is the equivalent of issuing a CESTPATH and CESTPAIR command for the PPRC pair. 3. Immediately suspend the new PPRC operation. This forces all changes made to the secondary to be recorded in the bit maps for the operation. The changes are not copied to the original primary, device number *ssss*. This is the equivalent of issuing a CSUSPEND command for the PPRC pair just established. 4. Redirect application I/Os from the primary volume to the secondary volume. Thus, all I/Os that were directed to the source device (*ssss*) are now directed to the target device (*tttt*). After completion of this option, device *ssss* is no longer involved in any operation.	Before this option is used, paths have been established in the opposite direction from the target device's storage control to the source device's storage control. In a **sysplex**, issue this reply from any system.

Terminate Swap (IGF520A, IGF521A, or IGF522A)	This is a request to end the swap operation.
	
	Message IGF512I message is issued to the system operator log as follows:
	`IGF512I SWAP FROM ssss TERMINATED - SWAP`
	`TERMINATED BY OPERATOR`
	
	P/DAS returns error code 16 to the system.
		In a **sysplex**, issue this reply from any system.

System actions for switch pair, and swap

When you reply to message IGF520A with "switch pair, and swap," P/DAS takes action that is based on the existing environment. Refer to Table 52 and to the actions labeled A through D to determine how P/DAS will respond to this reply.

Table 52. System actions for switch pair, and swap. Descriptions for the individual actions follow this table.

PPRC path direction	System action with equivalent device geometry	System action with different device geometry
PPRC paths exist in both directions	Action A	Action B
PPRC paths only exist from primary to secondary	Action C	Action D

Explanation of terms:

- **Paths exist in both directions** means that a path exists from the primary volume's storage control to the secondary volume's storage control, and another path exists from the secondary to the primary. The PPRC pair is able to operate in either direction, and the environment is ready to allow a switch of the PPRC pair from secondary to primary. When paths exist only from the primary to the secondary, the environment is not set up to allow data flow from the target volume to the source volume.
- **Equivalent device geometry** means that the primary and secondary volumes have equivalent track sizes and volume capacity.
- **Different device geometry** means that the primary and secondary volumes have the same track size but the secondary has a larger volume capacity than the primary.

The following specific actions relate to Table 52. The table refers to these actions according to your existing environment.

Action A:

There are PPRC paths in place from the primary volume to the secondary and also from the secondary to the primary. The source and target volumes have equivalent device geometry. In this situation, selecting "switch pair, and swap" directs P/DAS to perform the following actions:

1. End the current PPRC pair.
2. Establish a new PPRC pair (with NOCOPY) to activate copying in the reverse direction (from the target device *tttt* to the source device *ssss*).
3. Immediately suspend the PPRC operation. All changes made to target device *tttt* are recorded in the bit maps for the operation, but are not to be copied to the original source device *ssss*.
4. Redirect all application I/O to target device number *tttt*. The change is transparent to the application programs. P/DAS only sends I/Os to the secondary volume from this point on, and the primary and secondary volumes will no longer contain the same information.

The changes that occur between the swap operation and the subsequent resynchronization of the volume pairs are maintained on the old secondary volume. P/DAS will copy the changed updated tracks back to the original primary volume, and will make the volumes duplexed PPRC pairs again.

Action B:

There are PPRC paths in place from the primary volume to the secondary and also from the secondary to the primary. The target device *tttt* has greater volume capacity than the source device *ssss*. In this situation, your "switch pair, and swap" choice directs P/DAS to issue the following additional message to the system operator console:

```
IGF522A UNABLE TO SWITCH, FROM DEVICE IS SMALLER THAN TO DEVICE:
    REPLY 1 TO CONTINUE SWAP | 2 TO TERMINATE SWAP
```

Select one of the message options, as described in Table 51 on page 397.

If you choose to continue with the swap, the changes that occur between the swap and the subsequent resynchronization of the volume pairs will not be maintained on the old secondary volume.

Action C:

A PPRC path exists from the primary to the secondary, but not in the reverse direction. The primary and secondary volumes have the same physical capacity. In this situation, your “switch pair, and swap” choice directs P/DAS to issue the following additional message to the system operator:

```
  IGF521A NO PATH IN OPPOSITE DIRECTION: REPLY 1 TO CONTINUE SWAP |
     2 TRY AGAIN | 3 TERMINATE SWAP
```

Select one of the message options, as described in Table 51 on page 397.

Action D:

A PPRC path exists from the primary to the secondary, but not in the reverse direction. The primary and secondary volumes do not have the same physical capacity. P/DAS takes no special action to use the additional capacity on the target volume. In this situation, your “switch pair, and swap” choice directs P/DAS to issue the following additional message to the system operator:

```
IGF521A NO PATH IN OPPOSITE DIRECTION: REPLY 1 TO CONTINUE SWAP |
   2 TRY AGAIN | 3 TERMINATE SWAP
```

Select one of the message options, as described in Table 51 on page 397. If you reply with “2” (Try Again), P/DAS issues the following message, which requires an operator reply:

```
IGF522A UNABLE TO SWITCH, FROM DEVICE IS SMALLER THAN TO DEVICE:
   REPLY 1 TO CONTINUE SWAP | 2 TO TERMINATE SWAP
```

Select one of the message options, as described in Table 51 on page 397.

P/DAS error conditions

This section describes error conditions that can occur while attempting to perform P/DAS operations or during the resulting system actions.

Storage-related failures

The system obtains storage to perform P/DAS operations. If the storage requests fail, the system issues the following error message:

```
IGF512I SWAP FROM ssss TERMINATED - GETMAIN FAILED - SP245
```

P/DAS writes this message to the system operator log and returns error code 16 to the system.

I/O-related failures

The system issues I/O to the storage subsystems when PPRC pairs are establishing and ending as part of P/DAS functions. If a storage subsystem I/O fails during these functions, MVS issues the following error message:

```
IGF512I SWAP FROM ssss TERMINATED - SWAP FAILED DUE TO I/O ERROR
```

P/DAS writes this message to the system operator log and returns error code 16 to the system.

Validation failures

One of the following error messages are generated when a validation process fails. P/DAS writes all messages to the system operator log and returns error code 16 to the system.

- The following message means that the volumes are not part of a PPRC pair or that the volumes are not in duplexed mode:

```
IGF512I SWAP FROM ssss TERMINATED - NOT A VALID PPRC PAIR
```

- The following message means that the volume is an active paging volume:

```
IOS602I IOACTION - STOP NOT ALLOWED FOR PAGING DEVICES.
         THE FOLLOWING DEVICE(S) ARE IGNORED:  dev dev1-dev2...
```

- The following message means that a system in the sysplex has reserved the volume:

```
IOS604I IOACTION - STOP NOT ALLOWED FOR RESERVED OR RESERVED PENDING
         DEVICES.
         THE FOLLOWING DEVICE(S) ARE IGNORED:  dev dev1-dev2...
```

- The following message means that the volume is currently allocated:

```
IOS605I IOACTION - STOP NOT ALLOWED FOR DEVICES IN USE BY THE SYSTEM.
         THE FOLLOWING DEVICE(S) ARE IGNORED:  dev dev1-dev2...
```

If devices remain stopped

MVS issues the following message if devices remain stopped for an extended period of time:

```
IOS601I IOACTION - DEVICES REMAIN IN THE STOPPED STATE.
         USE THE 'D IOS,STOP' COMMAND TO DISPLAY THE DEVICES
```

P/DAS operations example

Table 53 shows the sequence of events that make up a P/DAS operation in a sysplex environment. As in all P/DAS operations, the correct command sequence is necessary to perform a successful P/DAS exchange of devices.

In this example, there are three systems in the sysplex that share volumes 191 and 581. System A is the main system and has the master console. Initialize the environment by establishing a single-direction PPRC path from the storage control for device 191 to the storage control for device 581. Device 191 is the primary volume of the PPRC pair, and device 581 is the secondary volume.

Note: If volume pairs are in PPRC extended distance mode, convert the pairs to synchronous mode to allow them to reach DUPLEX status before a swap operation begins. This ensures that the volume pairs are in sync and all updates are copied to the secondary.

In preparation for P/DAS operation, the storage administrator issues two CESTPAIR commands. The commands are issued:

- From any of the attached systems to establish a path from the storage control for device 191 to the storage control for device 581.
- From any of the attached systems for device 191 (primary) and device 581 (secondary). PPRC copies all of the tracks on device 191 onto device 581.

Table 53 shows what happens when the P/DAS operation begins.

Table 53. Example of a P/DAS swap function in a sysplex environment

System A (main system)	System B	System C
Operator issues: ROUTE *ALL,IOACTION STOP,DEV=191. System A stops I/O on 191, then issues IOS600I and IOS601I messages.	System B stops I/O on 191, then issues IOS600I and IOS601I messages.	System C stops I/O on 191, then issues IOS600I and IOS601I messages.
Operator issues: ROUTE *ALL,SWAP 191,581. System A performs validation, then issues IGF520A message.	System B performs validation, then issues IGF520A message.	System C performs validation, then issues IGF520A message.
Operator issues: R nn,1. System A ends PPRC pair, swaps pair, then issues IGF505I message.	After IGF505I message received, operator issues: R nn,3. System B swaps PPRC pair, then issues IGF505I message.	After IGF505I message received, operator issues: R nn,3. System C swaps PPRC pair, then issues IGF505I message.

Table 53. Example of a P/DAS swap function in a sysplex environment (continued)

System A (main system)	System B	System C
Operator issues: ROUTE *ALL,IOACTION RESUME, DEV=581. System A directs I/O to 581, then issues IOS607I message.	System B directs I/O to 581, then issues IOS607I message.	System C directs I/O to 581, then issues IOS607I message.

Part 4. Global Mirror

This provides the information to help you use global mirror.

Chapter 20. Planning for Global Mirror

Global Mirror combines PPRC-Extended Distance pairs (PPRC-XD) with FlashCopy consistency groups to provide true backup at a secondary site. This is accomplished through the addition of a master session in the hardware configuration. This master session controls all updates to the secondaries of the PPRC-XD pairs and, at the secondary site, the creation of consistent copies (using FlashCopy), at user specified intervals. Global Mirror for ESS requires FCP connection.

A Global Mirror session consists of a single master storage control, one or more optional, subordinate storage controls (up to the maximum allowed by microcode or 255, whichever is smaller), and one or more PPRC primary volumes. At the time a session is started, the master and parameters of the session are specified. The session should have been previously identified to LSSs that are going to participate in the session. These LSSs may reside in the master storage control or in subordinate storage controls. The session will be inactive until volumes are added to the session.

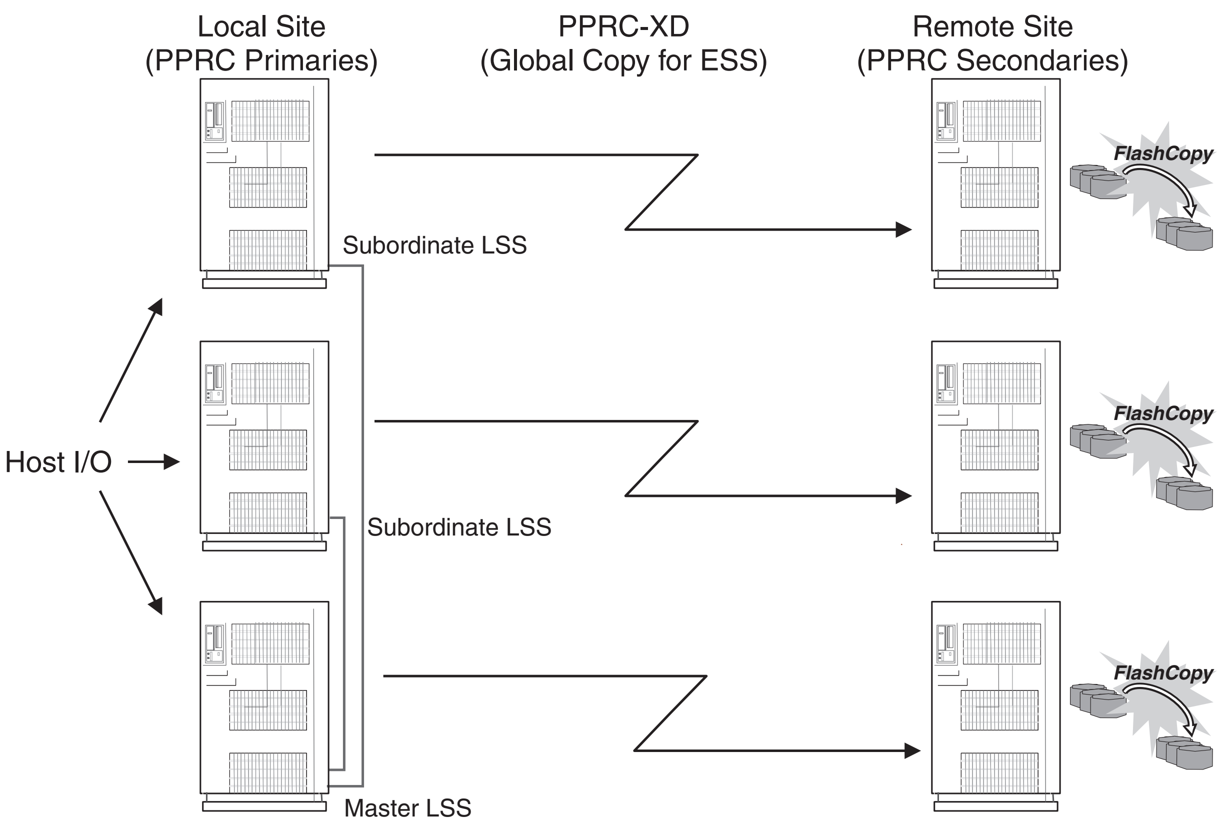


Figure 36. Global Mirror for ESS Combines PPRC-XD Pair with FlashCopy

Chapter 21, "Setting up Global Mirror for ESS," on page 411 explains how to configure the Global Mirror environment.

In this topic

The following sections are included in this topic:

Section . . .
"Global Mirror requirements"
"Global Mirror operational considerations" on page 409
"Controlling access to Global Mirror commands" on page 409
"Additional planning considerations" on page 410

Global Mirror requirements

You must meet the software and hardware requirements listed in the following sections in order to plan for and successfully use Global Mirror. Table 3 on page 15 includes additional information about PPRC hardware and software requirements.

Global Mirror software requirements

z/OS, VM, VSE, and ICKDSF [1] stand-alone systems support Global Mirror.

z/OS software requirements

ANTRQST API requests for Global Mirror functions are supported on all current releases of z/OS.

VM software requirements

VM/ESA 2.1.0 (and above) supports Global Mirror operations. See the items identified as "For VM Systems" under "PPRC operational considerations" on page 264.

ICKDSF support

ICKDSF release 17 and above supports Global Mirror operations for z/OS, VM, VSE, and stand-alone platforms. The ICKDSF PPRCOPY command provides functions that correspond to the PPRC MVS TSO command set.

For additional information about ICKDSF, refer to *Device Support Facilities (ICKDSF) User's Guide and Reference*.

Global Mirror hardware requirements

The following are PPRC hardware requirements:

- Both primary and recovery site storage controls must be of the same type (peer to peer).
- The PPRC V2 and FlashCopy V2 features must be enabled.
- Global Mirror requires Fibre Channel Protocol (FCP) connections between your primary and secondary subsystems. The FCP connections can be direct or through a fibre channel switch.
- Global Mirror also requires Fibre Channel Protocol (FCP) connections between the master storage control and each storage control subordinate in a session.
- PPRC-capable Licensed Internal Code (LIC) must be installed on both the primary and secondary storage subsystems. The levels of PPRC LIC on the systems must be compatible.

1. ICKDSF refers to the Global Mirror function as Asynchronous PPRCOPY.

- A compatible secondary volume must be available for each primary volume you need to copy. The secondary volume must have the identical track capacity and number of tracks per cylinder and either the same or larger volume capacity. Any volume in a simplex state, or one that has not completed its first pass copy, will remain in a join pending state and will not participate in the Global Mirror session.

Global Mirror operational considerations

Refer to "PPRC operational considerations" on page 264 for PPRC information. Chapter 26, "What is FlashCopy?," on page 471 contains information on operational considerations for FlashCopy.

Controlling access to Global Mirror commands

You can limit the use of Global Mirror commands by defining resource profiles in the RACF FACILITY class and restricting access to those profiles. To use a protected command, you need read-access authority to the applicable profile.

Table 54 lists the Global Mirror commands and the facility class profiles that can restrict them. See the *z/OS Security Server RACF Security Administrator's Guide* for details on activating the RACF facility class, and defining and authorizing users to the PPRC command profiles.

Table 54. Global Mirror FACILITY class profile names

Command	Profile Name
RQUERY	STGADMIN.ANT.PPRC.COMMANDS
RSESSION	STGADMIN.ANT.PPRC.COMMANDS
RVOLUME	STGADMIN.ANT.PPRC.COMMANDS
RQUERY	STGADMIN.ANT.PPRC.CQUERY

Note: Authorize RQUERY command use with the STGADMIN.ANT.PPRC.COMMANDS profile or the STGADMIN.ANT.PPRC.CQUERY profile. PPRC first checks STGADMIN.ANT.PPRC.COMMANDS for authorization. If authorization is not permitted with the STGADMIN.ANT.PPRC.COMMANDS profile, PPRC checks the STGADMIN.ANT.PPRC.CQUERY profile for authorization to issue the RQUERY command.

Examples: The following examples activate the RACF FACILITY class, define the profile for the PPRC commands, and give user STGADMIN authority to use this profile:

- This example activates the RACF FACILITY class:

```
SETROPTS CLASSACT(FACILITY)
```

- This example defines the profile for PPRC commands, and authorizes user STGADMIN to use this profile:

```
RDEFINE FACILITY STGADMIN.ANT.PPRC.COMMANDS UACC(NONE)

  PERMIT STGADMIN.ANT.PPRC.COMMANDS CLASS(FACILITY) -
         ID(STGADMIN) ACCESS(READ)
```

Additional planning considerations

See Chapter 12, “Planning for Peer-to-Peer Remote Copy,” on page 263 and Chapter 26, “What is FlashCopy?,” on page 471 for more specific information on PPRC and FlashCopy.

Chapter 21. Setting up Global Mirror for ESS

This topic describes how to set up the Global Mirror environment.

In this topic

The following sections are included in this topic:

Setting up PPRC and FlashCopy

In order to set up a Global Mirror session you must first establish PPRC-XD pairs. For information on PPRC-XD, refer to "Managing PPRC extended distance mode" on page 353. You must then establish the PPRC-XD secondaries as source in FlashCopy relationships, specifying the ASYNC MODE parameter. See "Subparameters for REQUEST=FCESTABLISH" on page 616.

If you withdraw an existing FlashCopy relationship in a Global Mirror session, the next consistency group will fail.

Establishing Global Mirror paths

To set up a Global Mirror environment, you must first establish a path between the master storage control and all subordinate storage controls participating in the session. Paths must also be established between the local LSSs that contain PPRC primaries and the remote LSSs that contain their PPRC-XD secondaries. It is not necessary to have paths from the master storage control to the secondaries unless the master storage control is participating in any PPRC-XD pairs. In that case, paths are required between the LSSs in the master storage control that contain PPRC primaries participating in the session and the LSSs that contain their PPRC secondaries. The master storage control and any subordinate storage controls must have local host connections so that commands can be issued to devices in the ESS where they reside.

ESCON links are not supported in Global Mirror environment for control paths or PPRC data paths. There must be at least one PPRC FCP link established between the master storage control and each of the subordinates when an RSESSION start or resume command is issued or else the command will fail. FCP paths are also required for the PPRC pairs that participate in the Global Mirror session. If ESCON paths are detected as the link between PPRC pairs in a session at the time the session is started, the start will fail. If ESCON paths are detected at any other time, for example when joining a volume to the session or establishing a PPRC pair for a simplex volume that has already been joined to a session, no error is received for the join or establish. In these cases, the next attempt to form a consistency group will fail. It is the responsibility of the issuer of the command to use either a polling mechanism (issue regular queries) or watch for any potential status change surfaced by the hardware to detect this problem.

Note:

1. Global Mirror only supports PPRC-XD pairs.
2. If you attempt to join a synchronous PPRC pair to a session, it will remain in join-pending state and never participate in a consistency group.
3. If you convert an existing PPRC-XD pair to synchronous PPRC pair, it will fail the next consistency group.
4. If you join a pair that does not have an appropriate FlashCopy relationship established, its status will remain as join pending.
5. If you are joining a multi-target environment, you must also identify the secondary so that Global Mirror mechanism knows which pair will participate in the session.

Steps for configuring a Global Mirror session

The recommended sequence of steps for setting up a global Mirror configuration is as follows.

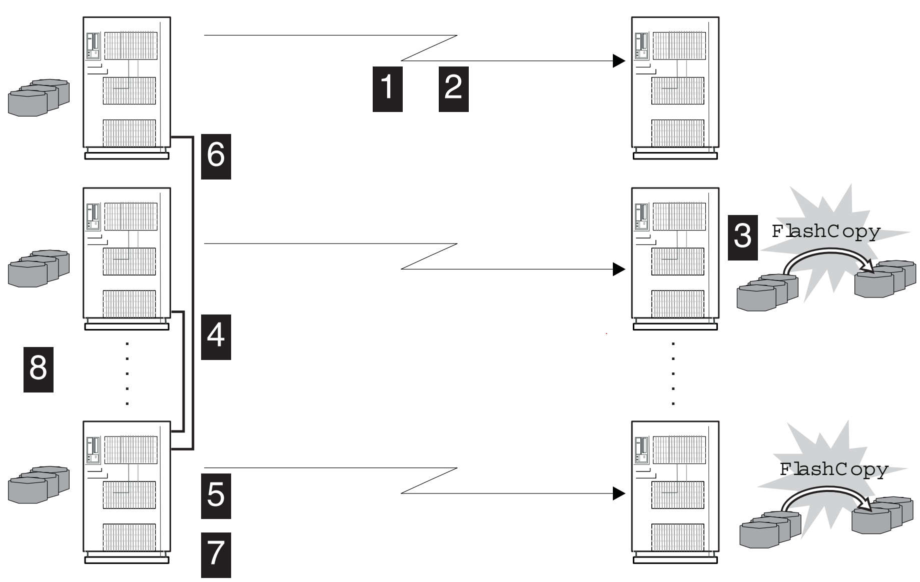


Figure 37. Setting Up Global Mirror for ESS

1 Use the PPRC Establish Path command to establish FCP paths between the LSSs that will contain the PPRC primaries and the LSSs that will contain the PPRC secondaries.

2 Use the PPRC Establish Pair command to establish PPRC-XD pairs that will participate in the session, specifying COPY in the MODE parameter.

3 Use the FlashCopy Establish command to establish FlashCopy relationships between the PPRC secondaries and their targets specifying ASYNC in the MODE parameter.

4 Use the PPRC Establish Path command to establish FCP paths between the intended master and intended subordinate ESSs that will be participating in the session.

5 Use the RSESSION DEFINE command to define the session (session ID created) to the LSSs in the master that will control the session.

6 Use the RSESSION DEFINE command to define the session to the LSSs in subordinate storage controls that are to participate in the session.

7 Issue an RSESSION START command to the master, specifying the topology and performance parameters for the session.

8 Use the RVOLUME JOIN command to populate the session with volumes.

Global Mirror session example

The following Assembler program is an example of how to start a Global Mirror session. The session will consist of a master storage control (serial number 0000000FC108) and three subordinate storage controls (serial numbers 000000018598, 000000018299, and 0000000FC132). The master storage control has LSSs corresponding to SSIDs B000 and B080. The subordinate storage controls have LSSs corresponding to SSIDs C100, 6000, and 8200. The GM session number is BC, and volume ZBOB40 is used as the access device

After coming back from ANTRQST, fields RTNCD and RSNCD are checked. If ANTRQST processing detects any syntax errors (for example, the action is specified as BEGIN instead of START) the error information is placed in RTNCD and RSNCD. If syntax checking is successful, the request is sent to address space ANTAS000 for processing. The first eight bytes of the return information field (RETINFO) contains the results of the execution.

```
TITLE 'RSESSION: Test ANTRQST REQUEST=RSESSION'
RSESSION CSECT ,
RSESSION AMODE 31
RSESSION RMODE ANY
         USING *,15
         B     @PROLOG
         DC    CL1' '
         DC    CL8'RSESSION'
         DC    CL1' '
         DC    CL8'&SYSDATC'
         DC    CL1' '
         DC    CL8'&SYSTIME'
         DC    CL1' '
         DROP  15
@PROLOG  STM   14,12,12(13)
         LR    12,15
@PSTART  EQU   RSESSION
         USING @PSTART,12
         L     0,@SIZDATD
         GETMAIN RU,LV=(0)
         LR    15,13
         LR    13,01
         USING @DATD,13
         ST    15,4(,13)
         ST    13,8(,15)
         LM    15,01,16(15)
         MVC   SNBR,=XL1'BC'
         MVC   VOLSER,=CL6'ZBOB40'
         MVC   ACTION,=CL12'START'
         MVC   MSSERIAL,=CL12'0000000FC108'
```

Figure 38. Global Mirror Session Example

```
*  Fill in subordinate storage controlinformation
*
         MVC   SBNBR,=XL1'03'
         MVC   SBMSSID1,=XL2'B000'
         MVC   SBSSSID1,=XL2'C100'
         MVC   SBSSERIAL1,=CL12'000000018598'
         MVC   SBMSSID2,=XL2'B080'
         MVC   SBSSSID2,=XL2'6000'
         MVC   SBSSERIAL2,=CL12'000000018299'
         MVC   SBMSSID3,=XL2'B080'
         MVC   SBSSSID3,=XL2'8200'
         MVC   SBSSERIAL3,=CL12'0000000FC132'
*
*  Invoke ANTRQST
*
         ANTRQST ILK=ESSRVCS,                                          X
               REQUEST=RSESSION,                                       X
               SNBR=SNBR,                                              X
               VOLSER=VOLSER,                                          X
               ACTION=ACTION,                                          X
               MSSERIAL=MSSERIAL,                                      X
               SBINFO=SBINFO,                                          X
               RETINFO=XRETINFO,                                       X
               RETCODE=RTNCD,                                          X
               RSNCODE=RSNCD,                                          X
               MF=(E,P_LIST)
         CLC   RTNCD,=F'0'          * Good return from syntax checking
         BE    SKIP                 * Yes
         MVC   RTCD,RTNCD           * No - save return code
         MVC   RSCD,RSNCD           *      save reason code
         B     ERROR
SKIP     DS    0H
         MVC   RTCD,RTC             * Get return code from RETINFO
         MVC   RSCD,RSC             * Get reason code from RETINFO
         CLC   RTCD,=F'0'           * Good return from execution
         BNE   ERROR                * No
*
*  Do REQUEST=RSESSION processing
*
         B     FINISH
ERROR    DS    0H
         CLC   RTCD,=A(RQST_PC_NUMBER_ZERO) * Have the API PC number?
         BNE   FINISH                * Yes
         WTO   'ANTAS000 NOT ACTIVE' * No
FINISH   DS    0H
         L     2,RTCD
         L     3,RSCD
         LR    1,13
         L     13,4(,13)
         LA    15,0
         L     0,@SIZDATD
         FREEMAIN RU,LV=(0),A=(1),SP=(15)
         LR    15,2
         LR    0,3
         L     14,12(,13)
         LM    1,12,24(13)
         BR    14
         LTORG
```

Figure 39. Global Mirror Session Example (continued)

```
@SIZDATD DS    0A
         DC    AL1(0)
         DC    AL3(@DYNSIZE)
@DATD    DSECT
         DS    18F
RTNCD    DS    F
RSNCD    DS    F
RTCD     DS    F
RSCD     DS    F
XRETINFO DS    0F,CL100
         ORG   XRETINFO
RTC      DS    F
RSC      DS    F
         ORG   XRETINFO+100
SNBR     DS    XL1
VOLSER   DS    CL6
ACTION   DS    CL12
MSSERIAL DS    CL12
subordinate storage control  DS    0CL49
SBNBR    DS    XL1
SBMSSID1 DS    XL2
SBSSSID1 DS    XL2
SBSSERIAL1 DS  CL12
SBMSSID2 DS    XL2
SBSSSID2 DS    XL2
SBSSERIAL2 DS  CL12
SBMSSID3 DS    XL2
SBSSSID3 DS    XL2
SBSSERIAL3 DS  CL12
         ANTRQSTL NAME=P_LIST,BASE=0F
         ORG   *+1-(*-@DATD)/(*-@DATD)
@ENDDATD DS    0X
@DYNSIZE EQU   ((@ENDDATD-@DATD+7)/8)*8
         END   RSESSION
```

Figure 40. Global Mirror Session Example (continued)

Chapter 22. Global Mirror command descriptions

This section describes Global Mirror TSO commands with their required and optional parameters, and includes an example for each command. Chapter 15, "Managing Peer-to-Peer Remote Copy operations," on page 321 contains other examples.

In this topic

The following sections are included in this topic:

Global Mirror command overview

Global Mirror TSO commands have the same restrictions as other TSO commands. The TSO abbreviation convention requires that you specify as much of the command parameter as is necessary to distinguish it from other parameters. Do not abbreviate Global Mirror command names.

Global Mirror TSO commands, like most TSO commands, are deferred requests. You receive an initial message that reports whether the command syntax is valid, and whether TSO has accepted the command for processing. Command function processing is a separate operation that can take minutes or hours to complete. The TSO terminal, meanwhile, is free to do other processing.

When TSO is operational, you can issue the Global Mirror commands to automatically initiate Global Mirror subsystem activity. You can put all Global Mirror TSO commands in a SYS1.PROCLIB member, in CLISTs, and include them within standard job JCL.

For additional information about TSO commands, refer to *z/OS TSO/E Command Reference*.

RQUERY – querying a Global Mirror session

RQUERY command syntax

The syntax of the RQUERY command is:

```
►►──RQUERY──SNBR──(──sessno──)──┬─DEVN(devno)─┬────────────────────┬─┬──ACTION(──┬─DVCSTAT─┬──)──────────►
                                │             └─SUBCHSET(subchset)─┘ │           ├─GMLSTAT─┤
                                └─VOLSER──(──volno──)────────────────┘           └─GMPSTAT─┘
```

```
►──┬────────────────────────────────────────────────────┬──────────────────────►◄
   └─LSSTYPE(──┬─FB──┬──────────────┬─DVCNBR(dvc)──┬──)─┘
               │     └─LSSNBR(lss)──┘              │
               └─CKD───────────────────────────────┘
```

Required parameters

SNBR A two-digit value ranging from 00 to FF specifying the Global Mirror session number for query. If this is not specified, every session number defined in the LSS where the query is directed is queried. If 00 is specified, all current sessions are queried.

VOLSER
Specifies the six-digit MVS volume number of the device to be used for I/O. When LSSTYPE(FB) is specified, this is a six-character MVS volume serial number of the device to be used as a CKD access device for I/O. This must be an online System z volume located in the same cluster as the LSS where the command is directed.

ACTION
Specifies the action to be performed:

DVCSTAT
Provides a formatted report to the TSO user with information pertaining to device specified in the request and the Global Mirror (or Asynchronous PPRC) session.

GMLSTAT
Provides a formatted report to the TSO user with summary information pertaining to the Global Mirror (or Asynchronous PPRC) session.

The control unit returns information for multiple sessions, with detail for sessions that were started in the same cluster as the device where the query was issued, and a status of NotOwnedCL, for sessions that were started in the other cluster, or NotMaster, for subordinate sessions. Information returned for sessions not owned by the cluster where the query was issued or for a subordinate session consists of:

- Session ID
- CG interval time
- Coordination interval time
- Max drain time
- Master CU sequence number.

If the IBM storage management software specified the SYSPLEXNAME keyword on START or RESUME, the following are included:

- Current sysplex clock time
- Drift
- Last Good CG sysplex clock time
- Sysplex name.

When multiple sessions are started for an SFI, ACTION=GMLSTAT with SNBR of 00 returns information for multiple sessions, separated by a line of asterisks (*). Specifying a SNBR other than

00 displays only query information for that session. The default is 00 (show all sessions in the query).

GMPSTAT
Provides a formatted report to the TSO user with summary information pertaining to the LSS specified in the request and the Global Mirror session.

LSSNBR
Specifies a two-character input field indicating the type of report to be generated. Valid values are hex characters in the range 00-FF. LSSNBR is required if LSSTYPE(FB) is specified, otherwise it defaults to the LSS where the device specified in the VOLSER keyword resides.

LSSTYPE
An optional three-character field identifying the type of devices in the LSS. Valid types are:

FB Open, or fixed block devices.

CKD Count key data, or System z-attached devices.

DVCNBR
A two-character field specifying the device for which the query is being issued. This is required when LSSTYPE is FB. When optionally specified for an LSSTYPE of CKD, this value is the cca.

DEVN
Specifies the four-digit MVS device number of the device to be used for I/O. When LSSTYPE(FB) is specified, this is a four-digit MVS device number of the device to be used as a CKD access device for I/O. This must be a System z device located in the same cluster as the LSS where the command is directed. **DEVN** is mutually exclusive with the **VOLSER**.

Optional Parameters

SUBCHSET
Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is supported by the host system's processor and configured for the device.

If you omit the parameter, the command is issued to the device that is logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

RQUERY command examples

The following are examples of the RQUERY command:

```
Query a specific Global Mirror session:
 RQUERY SNBR(24) VOLSER(11828) ACTION(GMLSTAT)

Query a specific CKD device:
 RQUERY SNBR(16) VOLSER(421652) ACTION(DVCSTAT)
    LSSTYPE(CKD)

Query specific LSS device:
 RQUERY SNBR(24) VOLSER(11828) ACTION(GMPSTAT)
    LSSNBR(32) LSSTYPE(FB) DVCNBR(00)
```

Refer to Chapter 23, “Querying Global Mirror,” on page 429 for sample output from the RQUERY command.

RSESSION – controlling a Global Mirror session

Use the RSESSION command to define and undefine the Global Mirror session id to all LSSs participating in the session. Start, stop, pause, or resume the session and specify the amount of time allowed for consistency group tasks.

The RSESSION START and RESUME commands may be issued to any CKD device and LSS in the master storage control. Once the Global Mirror session is active, all STOP, PAUSE, and RESUME commands to the master storage control must be issued to the same CKD LSS where the start was issued as long as the configuration exists. It is strongly recommended that the same device in the master storage control be used for all RSESSION commands. RQUERY output will show the CKD LSS where the Master was started.

If the RSESSION START or RESUME command is issued without consistency group interval, coordination interval, and coordination drain time specified, and the session is already active, the command is accepted as long as the session topology is specified with the subordinate storage controlparameter.

Once the Global Mirror session is started, you cannot issue an RSESSION START command to that session with different performance parameters specified for CGINTERVAL, CGDRAIN, or COORDINTERVAL. To change these parameters, you must first issue RSESSION PAUSE, then issue RSESSION RESUME with the new performance parameters. If you issue a START with different performance parameters to a session that is either active or paused, the new parameters are ignored. If you issue a RESUME without specifying any performance parameters, the default values are applied and any custom values previously specified with START are lost.

When an RSESSION START or RESUME command includes the CGINTERVAL, CGDRAIN, or COORDINTERVAL parameter, it must include all three. None of these three parameters can be specified without the others.

If the RSESSION START command is issued and a different asynchronous PPRC session is already active, the command will fail.

RSESSION command syntax

The syntax of the RSESSION command is:

```
►►──RSESSION──SNBR──(──session_number──)──┬─DEVN(devno)──┬────────────────────┬─┬── ACTION──────────────►
                                          │              └─SUBCHSET(subchset)─┘ │
                                          └─VOLSER──(──volno──)─────────────────┘
```

```
►─┤ options ├──────────────────────────────────────────────────────────────►◄
```

ACTION options:

```
├──┬─(──┬─DEFINE───┬─)─LSSTYPE(─┬─FB──┬─)─LSSNBR(lssnum)─ESSSERIAL(serno)──────┬──────┤
   │    └─UNDEFINE─┘            └─CKD─┘                                        │
   ├─(──┬─START──┬─)─┤ CG Options ├─MSSERIAL(serno)─┤ SBINFO ├─────────────────┤
   │    └─RESUME─┘                                                             │
   │                                           ┌─YES─┐          ┌─NO──┐         │
   └─(──┬─PAUSE───┬─)─MSSERIAL(serno)─MASTER(─┴─NO──┴─)─FORCE(─┴─YES─┴─)─┤ SBINFO ├┘
        ├─STOP────┤
        └─CGPAUSE─┘
```

CG Options:

```
├──┬─────────────────────────────────────────────────────────┬──────────────────────┤
   └─CGINTERVAL(secs)─CGDRAIN(secs)─COORDINTERVAL(msecs)─┘
```

SBINFO:

```
            ┌─,──────────────────────────┐
            ▼                            │
├──SBINFO(───mssid,─sssid,─serialno──┴─)──────────────────────────────────────┤
```

Required parameters

SNBR A two-character value ranging from 01 to FF specifying the Global Mirror session number.

VOLSER
Specifies the six-digit MVS volume number of the device to be used for I/O. When LSSTYPE(FB) is specified for ACTION(DEFINE) or ACTION(UNDEFINE), this is a six-character MVS volume serial number of the device to be used as a CKD access device for I/O. This must be an online System z volume located in the same cluster as the LSS where the command is directed.

DEVN
Specifies the four-digit MVS device number of the device to be used for I/O. When LSSTYPE(FB) is specified, this is a four-digit MVS device number of the device to be used as a CKD access device for I/O. This must be a System z device located in the same cluster as the LSS where the command is directed.

ACTION
Specifies the action to be performed:

DEFINE
Define a Global Mirror session to an LSS.

UNDEFINE
Remove the specified Global Mirror session.

START
Start the Global Mirror session. This identifies the master storage control, which will begin forming consistency groups for the specified global mirror session.

RESUME
Resume a paused Global Mirror session and optionally redefine the performance parameters for the session.

If the Global Mirror session was paused in a consistent manner using CGPAUSE, RESUME automatically unsuspends and resyncs the Global Copy pairs in the session.

PAUSE
Pause the specified Global Mirror session.

STOP Terminate the specified Global Mirror session.

CGPAUSE
Pause the Global Mirror session and suspend the Global Copy pairs in the session so that the secondaries are consistent.

LSSTYPE
This optional keyword identifies the type of LSS, either CKD or FB (fixed block). When a Global Mirror session is to be defined to a fixed block LSS, this parameter is required. LSSTYPE is ignored if specified on actions other than DEFINE or UNDEFINE. The default is CKD.

LSSNBR
This specifies the two-digit hexadecimal LSS number (00-FF) targeted by this command. This keyword is required for ACTION(UNDEFINE) and ACTION(DEFINE) when LSSTYPE(FB) is specified. This keyword is ignored for actions other than DEFINE or UNDEFINE.

ESSSERIAL
Required with DEFINE and UNDEFINE and ignored for all other actions, this specifies the 12-digit storage control number targeted by this command.

MASTER
MASTER(YES) specifies that this command is being issued to a master storage control. MASTER(NO) specifies that this is being issued to a subordinate storage control. The default is YES. This keyword is ignored for any action other than STOP.

MASTER(NO) and FORCE(YES) are mutually exclusive.

FORCE
Optional for STOP and ignored for all other actions. Valid values are NO (the default) and YES. Normally a Stop does not complete until a consistent copy of the data has been achieved. By specifying FORCE(YES) a Stop will be done even if a consistent copy of the data cannot be formed. YES may only be specified when issuing commands to the master. This is intended for use when there is an error that prevents a normal STOP from taking place. FORCE(YES) and MASTER(NO) are mutually exclusive.

MSSERIAL
Required for START, RESUME, PAUSE and STOP, and ignored for all other actions, this specifies the 12-digit serial number of the master storage control.

SBINFO
Required for START, RESUME, PAUSE and STOP, and ignored for all other actions, *sbinfo* identifies a list of ESS subordinate sets in the Global Mirror session. The values in a *set* come from the corresponding values that were used for the Establish Path command connecting the master storage control

and subordinate storage control. The SSIDs specified are for the control path between the master and the subordinates.

mssid Specifies the four-digit SSID number in the master storage control where the control path to this subordinate originates.

sssid Specifies the four-digit SSID number in the subordinate storage control where the control path from the master storage control terminates.

serialno
Specifies the twelve-digit serial number of the subordinate storage control.

Optional parameters

CGINTERVAL
Optional with START and RESUME and ignored for all other actions, this specifies, in seconds, the wait time between the end of one consistency group formation and beginning of the next. If 0 is specified, consistency groups are formed continuously. This parameter must be entered along with CGDRAIN and COORDINTERVAL. It cannot be entered alone.

CGDRAIN
Optional with START and RESUME and ignored for all other actions, this specifies, in seconds, the time allowed for updates on primaries to *drain* to remote secondaries. The default is 30 seconds. If 0 is specified the default is used. Valid values range from 0000 to FFFF. This parameter must be entered along with CGINTERVAL and COORDINTERVAL. It cannot be entered alone.

COORDINTERVAL
Optional with START and RESUME and ignored for all other actions, this specifies, in milliseconds, the maximum time allowed for the master storage control to coordinate a consistent data point with storage control subordinates. The default is 50 msecs. If 0 is specified the default is used. Valid values range from 0000 to FFFF. This parameter must be entered along with CGINTERVAL and CGDRAIN. It cannot be entered alone.

SUBCHSET
Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is supported by the host system's processor and configured for the device.

If you omit the parameter, the command is issued to the device that is logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

RSESSION command examples

The following are examples of the RSESSION command:

```
Define an FB (open device) session:
 RSESSION ACTION(DEFINE) VOLSER(PAN001) SNBR(40) LSSNBR(2A)
     ESSSERIAL(22243) LSSTYPE(FB)

Start the session:
 RSESSION ACTION(START) VOLSER(PAN001) SNBR(40) MSSERIAL(22243)
     CGINTERVAL(30) CGDRAIN(45) COORDINTERVAL(60)
     SBINFO(FF23,FF25,18598,FF23,FF27,18299)

Pause the session:
 RSESSION ACTION(PAUSE) VOLSER(PAN001) SNBR(40) MSSERIAL(22243)
     SBINFO(FF23,FF25,18598,FF23,FF27,18299)

Resume the session:
 RSESSION ACTION(RESUME) VOLSER(PAN001) SNBR(40) MSSERIAL(22243)
     CGINTERVAL(45) CGDRAIN(60) COORDINTERVAL(90)
     SBINFO(FF23,FF25,18598,FF23,FF27,18299)
```

RVOLUME – manage volumes for Global Mirror session

Use the RVOLUME command to add and remove volumes for a Global Mirror session.

RVOLUME command syntax

The syntax of the RVOLUME command is:

```
>>--RVOLUME--SNBR--(--session_number--)--+-DEVN(devno)--+-------------------------+-+--LSSTYPE(--+-FB--+--)------->
                                         |               '-SUBCHSET(subchset)-----' |            '-CKD-'
                                         '-VOLSER--(--volno--)----------------------'

>--LSSNBR(lssnum)--ESSSERIAL(serno)--ACTION----(--+-JOIN---+--)------------------------------------------------>
                                                  '-REMOVE-'

        .-------.
        V       |
>--+-VOLLIST(----cca,---)-----------------------------------+------------------------------------------------><
   |          .-----------.                                 |
   |          V           |                                 |
   +-VOLRANGE(----cca1-cca2,---)----------------------------+
   |           .------------------------------.             |
   |           V                              |             |
   +-MTVOLLIST(----pcca,secser#,slss,scca-----)-------------+
   |            .------------------------------------.      |
   |            V                                    |      |
   '-MTVOLRANGE(----ppca1:ppcax,secser#,slss,scca1-----)-----'
```

Required parameters

SNBR A two-digit value ranging from 01 to FF specifying the Global Mirror session number.

VOLSER
Specifies the six-digit MVS volume number of the device to be used for I/O. When LSSTYPE(FB) is specified, this is a six-character MVS volume

serial number of the device to be used as a CKD access device for I/O. This must be an online System z volume located in the same cluster as the LSS where the command is directed. **VOLSER** is mutually exclusive with **DEVN**.

LSSTYPE
Specifies the type of LSS, FB or CKD. If the device(s) being joined or removed from the global mirror session are fixed block, LSSTYPE(FB) must be specified. CKD is the default.

LSSNBR
Specifies the two-digit hexadecimal LSS number (00-FF) targeted by this command. For CKD devices, this must be the same LSS where the volume specified by the VOLSER keyword resides.

ESSSERIAL
Specifies the 12-digit storage control number targeted by this command.

ACTION
Specifies the action to be performed:

JOIN Add device(s) to Global Mirror session.

REMOVE
Remove device(s) from Global Mirror session.

VOLLIST
Specifies a list of two-digit CCA (CKD device) or LUN (Open device) numbers to be added to or removed from the Global Mirror session. Up to 256 cca or lun numbers may be specified, separated by commas. This parameter is mutually exclusive with VOLRANGE, MTVOLLIST or MTVOLRANGE.

VOLRANGE
Specifies a list of two-digit CCA (CKD device) or LUN (Open device) ranges to be added to or removed from the Global Mirror session. The first cca or lun is the beginning of the range, and the second, separated by a colon, is the end. Up to 256 cca or lun pairs may be specified, separated by commas. This parameter is mutually exclusive with VOLLIST, MTVOLLIST or MTVOLRANGE.

MTVOLLIST
Specifies a list of Global Copy device numbers to be added to or removed from the Global Mirror session.

pcca Specifies the primary CCA (or LUN)

secser# Specifies the serial number of the secondary device's SFI for the Global Copy pair

slss Specifies the LSS number of the secondary device for the Global Copy pair

scca Specifies the CCA (or LUN) number of the secondary device for the Global Copy pair

Specify up to 255 device sets. Use a comma to separate the values in a set, and use a blank to separate sets, for example: `MTVOLLIST(C4,000000016191,3A,C4 C5,000000016191,3A,C5)`. This parameter is mutually exclusive with MTVOLRANGE, VOLLIST and VOLRANGE. When there are multiple secondaries, you must use MTVOLLIST or MTVOLRANGE.

MTVOLRANGE
Specifies a list of primary device ranges in an LSS to be added to or removed from the Global Mirror session.

pcca1:pccax
Specifies the first and last primary CCA (or LUN) in a range of contiguous primary devices. The first and last CCA indicated are separated by a colon.

secser# Specifies the serial number of the secondary device's SFI for the Global Copy pair

slss Specifies the LSS number of the secondary device for the Global Copy pair

scca1 Specifies the first CCA (or LUN) of the secondary device for the Global Copy pair. The last secondary CCA is derived from the number of devices specified in the *pcca1-pccax* range.

Specify up to 255 device sets. Use a comma to separate the values in a set, and use a blank to separate sets, for example: `MTVOLRANGE(C4:CF,000000016191,3A,C4 A0:BF,000000016191,30,A0 E0:E4,000000017720,D0,E0)`. This parameter is mutually exclusive with MTVOLLIST, VOLLIST, and VOLRANGE. When there are multiple secondaries, you must use MTVOLLIST or MTVOLRANGE.

DEVN
Specifies the 4-digit hexadecimal address of the device to which the I/O operation is directed. **DEVN** is mutually exclusive with **VOLSER**.

Optional parameters

SUBCHSET
Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is supported by the host system's processor and configured for the device.

If you omit the parameter, the command is issued to the device that is logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

RVOLUME command examples

The following are examples of the RVOLUME command:

1. Join volumes to a session:
```
RVOLUME SNBR(16) VOLSER(421652) LSSTYPE(FB) LSSNBR(2A) ESSSERIAL(51214)
     ACTION(JOIN) VOLLIST(C4,C5,C6,CA,CB)
```
2. Remove volumes from a session:
```
RVOLUME SNBR(16) VOLSER(421652) LSSTYPE(FB) LSSNBR(2A) ESSSERIAL(51214)
     ACTION(REMOVE) VOLRANGE(C4:C6,CA:CB)
```
3. Join a list of Global Copy devices to the Global Mirror session, in a multi-target environment:
```
RVOLUME SNBR(16) VOLSER(421652) LSSTYPE(FB) LSSNBR(2A) ESSSERIAL(000000051214) ACTION(JOIN)
    MTVOLLIST(C4,000000016191,3A,C4 C5,000000016191,3A,C5 C6,000000017720,3A,F6)
```
4. Join a list of ranges of Global Copy devices to the Global Mirror session, in a multi-target environment:

```
RVOLUME SNBR(16) DEVN(0F70) LSSTYPE(FB) LSSNBR(2A) ESSSERIAL(000000051214) ACTION(JOIN)
   MTVOLRANGE(C4:CF,000000016191,3A,C4 A0:BF,000000016191,30,A0 E0:E4,000000017720,D0,E0)
```

Chapter 23. Querying Global Mirror

In this topic

The following sections are included in this topic:

Deciding which query to use

The RQUERY API request returns unformatted information or formatted information at a number of different configuration levels, depending on which ACTION parameter is specified. The RQUERY API request uses ANTRQMAP DSECT to output unformatted query results.

Table 55. RQUERY ACTION Parameter Outputs

ACTION parameter specified	API	TSO	Level of query	Type of information	Formatted or Unformatted
STAT4ALSS	X		All sessions within LSS	Performance counters, consistency group information, FlashCopy information, and out-of-sync tracks	Unformatted
STAT4AESS	X		All sessions within ESS		
STAT4ACGRP	X		All sessions within ESS		
STAT4BLSS	X		All sessions within LSS	Out-of-sync information	
STAT4BESS	X		All sessions within ESS		
STAT4C	X		All sessions within ESS	Summary information	
STAT51	X		All Global Mirror sessions	Detailed performance information	
GMLSTAT	X	X	Global Mirror session	Summary information	Formatted
GMPSTAT	X	X	All sessions within LSS	Summary information	
DVCSTAT	X	X	Device	Summary information	

The following topics provide examples of RQUERY API requests and describe the formatted output they generate.

Querying Global Mirror summary output at session level

The RQUERY GMLSTAT request queries a Global Session session for master storage control information, volume count, consistency group revertible status, out-of-sync track count, and session control information.

```
RQUERY  LSSNBR(0D) ACTION(GMLSTAT) VOLSER(MAN001) DVCNBR(03) LSSTYPE(CKD)
ANTP7000I RQUERY Output Volser(MAN001) Action(GMLSTAT) Version(003)
ANTP7012I SNbr GMLStat    GoodCg   Pct CrnBadCG TotBadCG LastGoodCGSCntlClock
ANTP7002I --   ---------- -------- --- -------- -------- --------------------
ANTP7013I CD   NotOwnedCL
ANTP7003I .                                                                  .
ANTP7014I Master: Serial       SSID LSS   CGInt CGDrn CrdInt
ANTP7002I         ------------ ---- --    ----- ----- -----
ANTP7015I         0001075PK311 240E 0E        0    30    50
ANTP7003I .                                                                  .
ANTP7016I Subordinate: Serial       SSID LSS MSSID  Serial       SSID LSS MSSID
ANTP7002I              ------------ ---- --  ----   ------------ ---- --  ----
ANTP7017I              0001075PH971 680A 0A  240E
ANTP7003I .                                                                  .
ANTP7022I CurrentSCntlClock     CurrentGMTClock
ANTP7002I --------------------  --------------------
ANTP7023I No Time Available     14 May 2010 22:29:28
ANTP7003I .                                                                  .
ANTP7006I ********************************************************************
ANTP7003I .                                                                  .
ANTP7012I SNbr GMLStat    GoodCg   Pct CrnBadCG TotBadCG LastGoodCGSCntlClock
ANTP7002I --   ---------- -------- --- -------- -------- --------------------
ANTP7013I CC   Running    000000F3 100 00000000 00000000 14 May 2010 22:39:32
ANTP7003I .                                                                  .
ANTP7014I Master: Serial       SSID LSS   CGInt CGDrn CrdInt
ANTP7002I         ------------ ---- --    ----- ----- -----
ANTP7015I         0001075PK311 240D 0D        0    30    50
ANTP7003I .                                                                  .
ANTP7016I Subordinate: Serial       SSID LSS MSSID  Serial       SSID LSS MSSID
ANTP7002I              ------------ ---- --  ----   ------------ ---- --  ----
ANTP7017I              0001075PH971 6809 09  240D
ANTP7003I .                                                                  .
ANTP7022I CurrentSCntlClock     CurrentGMTClock
ANTP7002I --------------------  --------------------
ANTP7023I 14 May 2010 22:39:32  14 May 2010 22:29:28
ANTP7003I .                                                                  .
ANTP7006I ********************************************************************
ANTP7003I .                                                                  .
ANTP7012I SNbr GMLStat    GoodCg   Pct CrnBadCG TotBadCG LastGoodCGSCntlClock
ANTP7002I --   ---------- -------- --- -------- -------- --------------------
ANTP7013I CE   NotMaster
ANTP7003I .                                                                  .
ANTP7014I Master: Serial       SSID LSS   CGInt CGDrn CrdInt
ANTP7002I         ------------ ---- --    ----- ----- -----
ANTP7015I         0001075PH971
ANTP7003I .                                                                  .
ANTP7006I ********************************************************************
ANTP7003I .                                                                  .
ANTP7012I SNbr GMLStat    GoodCg   Pct CrnBadCG TotBadCG LastGoodCGSCntlClock
ANTP7002I --   ---------- -------- --- -------- -------- --------------------
ANTP7013I CF   NotMaster
ANTP7003I .                                                                  .
ANTP7014I Master: Serial       SSID LSS   CGInt CGDrn CrdInt
ANTP7002I         ------------ ---- --    ----- ----- -----
ANTP7015I         0001075PH971
ANTP7003I .                                                                  .
ANTP7006I ********************************************************************
```

When SNBR is not specified in the RQUERY GMLSTAT command, information is returned about any session that is defined in that LSS.

The fields in RQUERY GMLSTAT formatted output are defined as follows:

Table 56. GMLSTAT Field Definitions

Header/Label	Valid Value(s)	Description
SNbr	Hex value X'01' - X'FF'	Global Mirror session number
GMLStat	10 characters	Global Mirror session status
	CGCorruptd The Global Mirror session is in a fatal state. The reason for the fatal state is, the consistency group is corrupted. **CGPaused** The Global Mirror session was paused; consistency was specified. **CGPausing** The Global Mirror session is in the process of pausing; consistency was specified. **CmnFailure** The Global Mirror session is in a fatal state. The reason for the fatal state is, a communications failure has occurred during consistency check. **ConChkBusy** The Global Mirror session is in a fatal state. The reason for the fatal state is, a busy condition prevents consistency check from occurring. **ConChkFail** The Global Mirror session is in a fatal state. The reason for the fatal state is, the Global Mirror master was unable to complete consistency check. **ConChkTimO** The Global Mirror session is in a fatal state. The reason for the fatal state is, a timeout occurred during consistency check. **InvldSesId** The Global Mirror session is in a fatal state. The reason for the fatal state is, a Global Mirror invalid session Id was detected by the master. **NoConfig** There is no master or subordinate global mirror configuration defined for the specified session number in the cluster where the query was issued. When multiple sessions are started for an SFI, ACTION STAT51 with SNBR of 00 returns information for multiple sessions. The existing ANTRQMAP array is populated with that returned information. Specifying a SNBR other than 00 displays only query information for that session. The default is 00 (return all sessions). **NotMaster** The query was issued to a control unit that is not the master, therefore session status in not reported. **NotOwnedCL** The session is not owned by the cluster receiving the query. Only static information is available from the non-owning cluster. To get detail information for this session, issue the RQUERY to a device in the owning cluster. **Paused** The Global Mirror session has been paused. **Pausing** The Global Mirror session is in the process of pausing.	

Table 56. GMLSTAT Field Definitions (continued)

Header/Label	Valid Value(s)	Description
GMLStat (continued)	**Recovering** The SFI is in a recovery process (triggered, for example, by an IML, warmstart or failover/failback events). Only static information is available for sessions when recovery processes are in progress. **Running** The Global Mirror session is running. **RvtFcFaild** The GM session is in a fatal state. The reason for the fatal state is, the revert FlashCopy command, issued internally, failed. **RvtTimeOut** The GM session is in a fatal state. The reason for the fatal state is, FlashCopy revert failed due to a timeout. **SeqNbrBad** The GM session is in a fatal state. The reason for the fatal state is, the GM consistency check of sequence numbers failed. **StrucInacs** The GM session is in a fatal state. The reason for the fatal state is, the GM (internal) structures are inaccessible. **SubordTerm** The GM session is in a fatal state. The reason for the fatal state is, a subordinate has been terminated by a pause or stop request . **UnAvailbl** The storage control is unable to access internal control blocks that maintain status information. The session number (SNbr) will be reflected as '00'. **UNSFailed** The Global Mirror session was paused because the un-suspend failed.	
GoodCG	Hex value X'00000000' - X'FFFFFFFF'	The number of good consistency groups that have been formed since the Global Mirror session was started.
Pct	Decimal value between 0 - 100	Percentage of attempts to form consistency groups that were successful since the Global Mirror session was started.
CrnBadCG	Hex value X'00000000' - X'FFFFFFFF'	Current consecutive number of attempts to form a consistency group that have failed. Once a consistency group has been successfully formed this value is reset to zero.
TotBadCG	Hex value X'00000000' - X'FFFFFFFF'	Total number of attempts to form a consistency group that have failed since the Global Mirror session started
LastGoodCGSCntlClock	Character String	The value from the storage control internal clock when the last successful consistency group was formed.
CurrentSCntlClock	Character String	The value from the storage control internal clock at the time the query was issued.
CurrentGMTClock	Character String	The value from the system GMT timestamp at the time the query was issued.
Master: This section reports information pertinent to the Master storage control for the Global Mirror session.		
Serial	12-character ESS serial number	Serial number of the Master storage control for the Global Mirror session.

Table 56. GMLSTAT Field Definitions (continued)

Header/Label	Valid Value(s)	Description
SSID	Hex value X'0000' - X'FFFF'	SSID for the LSS in the Master storage control for the Global Mirror session where the start was issued. If the global mirror session contains only FB LSSes, the SSID shown here will be the SSID for the CKD access device where the start command was issued.
LSS	Hex value X'00' - X'FF'	LSS number in the Master storage control for the Global Mirror session where the start was issued.
CGint	Seconds - decimal value	Consistency group formation interval specified on session start or resume.
CGDrn	Seconds - decimal value	Maximum consistency group drain time specified on session start or resume.
CrdInt	Milliseconds - decimal value	Maximum consistency group coordination interval specified on session start or resume.
Subordinate: This section contains 2 columns of subordinate information. Lines are repeated as needed to report each subordinate storage control and their LSSes that are participating in the Global Mirror session.		
Serial	12-character ESS serial number	Serial number of the subordinate storage control for the Global Mirror session.
SSID	Hex value X'0000' - X'FFFF'	SSID for the LSS in the subordinate storage control that is 'connected' to the master. This is the SSID for the subordinate that was specified in subordinate storage controlon the start for the Global Mirror session.
LSS	Hex value X'00' - X'FF'	LSS number in the subordinate storage control participating in the Global Mirror session.
MSSID	Hex value X'0000' - X'FFFF'	SSID for the LSS in the master storage control that is 'connected' to the subordinate. This is the SSID for the master that was specified in subordinate storage controlon the start for the Global Mirror session.
BadCGrpFormation: Bad consistency group formation instances are reported in this section. These errors are due to internal commands issued by the control unit or between control units during consistency group formation, so much of the information provided is for diagnostic purposes only and may have to be interpreted by hardware support personnel.		
When		
	LAST	Information about the last consistency group formation error.
	PREV	Information about the next to last consistency group formation error.
	FIRST	Information about the first consistency group formation error.
Serial	12-character ESS serial number	Serial number of the storage control where the consistency group formation error occurred.
SSID	Hex value X'0000' - X'FFFF'	SSID for the LSS where the consistency group formation error occurred. If the SSID cannot be determined, or the failure did not occur in a specific LSS, this field will show '????'. This can happen if there is a communications failure between the master and the subordinate, or if a subordinate is stopped while a session is running.

Table 56. GMLSTAT Field Definitions (continued)

Header/Label	Valid Value(s)	Description
LSS	Hex value X'00' - X'FF'	LSS where the consistency group formation error occurred. A valid value for LSS is always displayed, but if the LSS cannot be determined, or the failure did not occur in a specific LSS, this field will show 'FF'. This can happen if there is a communications failure between the master and the subordinate, or if a subordinate is stopped while a session is running.
Reason	1. StrucInacs 2. CommPathFl 3. BadState 4. DrTimeExcd 5. InvaldParm 6. TempUnaval 7. LongBusy 8. CmdRej *xxx* 9. MCdErr *xxx*	1. The Master storage control is unable to access its internal structures 2. A communication path failure occurred 3. The session is in an invalid state 4. Maximum consistency group drain time exceeded 5. An invalid parameter was received by the storage control (this is an internal error) 6. Internal resources are temporarily unavailable 7. A long busy was received 8. An internally generated storage subsystem command was rejected. The last three characters (*xxx*) of the reason code represent the message and reason associated with rejection of a Format 0 command. These errors are not expected by the software and must be reported to IBM and interpreted by hardware personnel. Any other Format command error is reported as MCdErr. 9. A storage subsystem microcode error has occurred. If consistency group formation is failing repeatedly, contact the IBM support center to report a hardware problem and provide the last three characters (*xxx*) of the reason code.
Activity (the state of the master at the time of the consistency group failure)	1. Pausing 2. Starting 3. BtwnCGForm 4. StrtIncr 5. RunInPrg 6. DrnInPrg 7. EstabRvt 8. WithCommit 9. IncrCmplt 10. WithRevert 11. Paused 12. PostCGTsks 13. Fatal 14. ErrRecovry	1. A PAUSE or STOP Global Mirror is in progress 2. A START or RESUME Global Mirror is in progress 3. Global Mirror between consistency group formations 4. A Global Mirror start increment, or consistency group is in progress 5. A Global Mirror RUN is in progress 6. A Global Mirror drain is in progress 7. An Establish FlashCopy with Revertible is in progress 8. A Withdraw FlashCopy with Commit is in progress 9. A Global Mirror Increment Complete is in progress 10. A Withdraw FlashCopy with Revert is in progress 11. The Global Mirror is Paused 12. Global Mirror is performing post-consistency group tasks 13. Global Mirror is in a Fatal state 14. Global Mirror is completing error recovery

Querying Global Mirror summary output at LSS level

The RQUERY GMPSTAT request queries a Global Session in an LSS for volume count, out-of-sync track count, and session control information.

The following example screen shows the output that is received from the following command:

```
RQUERY ACTION(GMPSTAT) DEVN(F92)
```

```
RQUERY SNBR(BC) DEVN(0F92) ACTION(GMPSTAT) LSSNBR(07)

11 Oct 2004  09:41:56.03
ANTP7000I RQUERY Output Devn(0F92) SCHSET(0)  Action(GMPSTAT) Version(002)
ANTP7004I SNbr LSS GMPStatus  TotVol OOSVol OOSTrks
ANTP7002I --   --  ---------- ------ ------ --------
ANTP7005I 01   02  CGInPrgrs       1      0 00000000
```

Note: Do not use ANTRQST STAT4AESS and STAT4BESS requests in production environments. These requests can cause Global Mirror consistency groups to fail.

When SNBR is not specified in the RQUERY GMPSTAT command, information is returned about any session that is defined on that LSS.

The fields in RQUERY GMPSTAT formatted output are defined as follows:

Table 57. GMPSTAT Field Definitions

Header/Label	Valid Value(s)	Description
SNbr	Hex value X'01' - X'FF'	Global Mirror session number
LSS	Hex value X'01' - X'FF'	LSS number. If there are no volumes 'joined' to the session in the LSS that is being queried, this field will contain blanks.

Table 57. GMPSTAT Field Definitions (continued)

Header/Label	Valid Value(s)	Description
GMPStatus	10 characters 1. WrongState 2. UnAvailbl 3. NotStarted 4. NoVolumes 5. NoSessions 6. NoVolsActv 7. Normal 8. IncrPndg 9. CGInPrgrs 10. NoChngRcrd 11. ESCON 12. NotReady 13. VolUnAvail	Global Mirror session status 1. Session is either timed out or in another state preventing a consistency group from being formed. 2. Data could not be collected due to the unavailability of internal storage control structures (control blocks). The values for TotVol, OOSVol, and OOSTrks may be unavailable or unreliable. 3. Data could not be collected because the Global Mirror session has not been started with this LSS and subordinate in the topology. 4. There are no volumes in the session. 5. No Global Mirror session has been defined to this LSS. 6. There are no pending or active volumes in this LSS in the Global Mirror session . 7. The Global Mirror session is active and processing normally. 8. The Global Mirror session is in an increment pending state. 9. The global Mirror session is in the process of forming a consistency group. 10. Change recording is not active on the FlashCopy relationships. 11. ESCON paths detected. 12. One or more volumes are not in a state to allow a consistency group to be formed. 13. One or more volumes in the Global Mirror session is inaccessible, offline, in a "status cannot be determined state" or in CE mode. The values for TotVol, OOSVol, and OOSTrks may be unavailable or unreliable.
TotVol	Six-digit decimal number between 0-999999	Number of PPRC primary volumes in joined status in the specified session in this LSS.
OOSVol	Six-digit decimal number between 0-999999	Number of out of sync volumes in the specified session in this LSS.
OOSTrks	Hex value X'00000000' - X'FFFFFFFF'	Number of out of sync tracks.

Querying Global Mirror summary output at device level

The RQUERY DVCSTAT request queries a Global Mirror session for volume count and consistency group revertible status information at the device level. The following example is for a cascaded (non-multi-target) environment.

```
RQUERY SNBR(BC) VOLSER(ZBOB60) ACTION(DVCSTAT)

11 Oct 2004  09:41:03.73

ANTP7000I RQUERY Output Volser(ZBOB60) Action(DVCSTAT) Version(001)
ANTP7008I SNbr LSS Dvc VolStat              PriPPRCStat SecCascStat
ANTP7002I --   --  --  -------------------- ----------  ----------
ANTP7009I BC   07  09  InSession            DplxPendng  Simplex
ANTP7009I          15  InSession            DplxPendng  Simplex
```

When a specific, non-zero SNBR is specified in the RQUERY DVCSTAT command, the data lines in the summary output are repeated for each device in the requested session in the LSS. When SNBR is specified as zero, the data lines are repeated for each device in all currently defined sessions in the LSS when the command was issued.

In a multi-target environment, an additional ANTP7026I message is inserted for each device in the session, to display the secondary serial, LSS, and CCA, as follows:

```
ANTP7000I RQUERY Output Devn(0F51) SCHSET(0) Action(DVCSTAT) Version(002)
ANTP7008I SNbr LSS Dvc VolStat              PriPPRCStat SecCascStat
ANTP7002I --   --  --  -------------------- ----------  ----------
ANTP7009I 33   30  00  InSession            DplxPendng  Simplex
ANTP7026I             SECINFO: SERIAL: 0000000CRB71 LSS: 30 CCA: 0A
ANTP7009I      30  01  InSession            DplxPendng  Simplex
ANTP7026I             SECINFO: SERIAL: 0000000CRB31 LSS: 30 CCA: 01
ANTP7009I      30  02  InSession            DplxPendng  Simplex
ANTP7026I             SECINFO: SERIAL: 0000000CRB31 LSS: 30 CCA: 02
```

If MTVOLLIST or MTVOLRANGE was not specified on the RVOLUME JOIN command, the values in the ANTP7026I messages are periods (.), for example:

```
ANTP7026I             SECINFO: SERIAL: ............ LSS: .. CCA: ..
```

The fields in RQUERY DVCSTAT formatted output are defined as follows:

Table 58. DVCSTAT Field Definitions

Header/Label	Valid Value(s)	Description
SNbr	Hex value X'01' - X'FF'	Global Mirror session number
LSS	Hex value X'01' - X'FF'	LSS number.
Dvc	Hex value X'01' - X'FF'	Device number. For a CKD device, this will be the Channel Connection Address (CCA). For a Fixed Block (FB) device, this will be a Logical Unit Number (LUN).

Table 58. DVCSTAT Field Definitions (continued)

Header/Label	Valid Value(s)	Description
VolStat	10 character string, combining status values reported by the storage control, separated by dashes (-). 1. InSession|In 2. NotInSession|NIn 3. CascOK 4. JoinP 5. RemvP 6. 1st 7. Unavailabl 8. NoSessions 9. NoVolumes	Global Mirror session status 1. The volume is in a Global Mirror session. If no additional status is to be reported (cascading OK, remove pending, etc.), this will be displayed as 'InSession'. If additional status is to be reported, this will be displayed abbreviated as 'In'. 2. The volume is not in a session. If no additional status is to be reported (join pending, 1st pass copy in progress, etc.), this will be displayed as 'NotInSession'. If additional status is to be reported, this will be displayed abbreviated as 'NIn'. 3. The PPRC primary was established with CASCADE=YES. 4. The PPRC pair has not fully joined the ASYNC session. 5. The PPRC pair is in the process of being removed from the ASYNC session. 6. The PPRC pair is in its initial (first pass) copy phase. 7. Data could not be collected due to the unavailability of internal storage control structures (control blocks). 8. No Global Mirror session has been defined to the LSS where the requested device resides. 9. No volumes have been joined to the global mirror session.
PriPPRCStat	10-Character string 1. Simplex 2. DplxPending 3. Duplex 4. Suspended	The PPRC status for the primaries of the PPRC pairs in this LSS in the Global Mirror session: 1. The device is simplex. 2. The pair is in a duplex pending state. 3. The pair is in full duplex state. 4. The pair is suspended.
SecCascStat	10-Character string 1. Simplex 2. DplxPending 3. Duplex 4. Suspended	Status of the device as a cascaded secondary of a PPRC pair: 1. The device is not participating in a cascaded PPRC relationship. 2. The cascaded pair is in duplex pending state. 3. The cascaded pair is in full duplex state. 4. The cascaded pair is suspended.
SECINFO	SERIAL, LSS and CCA	**SERIAL** Serial number of the secondary for the relationship participating in the Global Mirror session. If MTVOLLIST or MTVOLRANGE was not specified on the RVOLUME JOIN command, the serial number is displayed as a string of periods (.). **LSS** LSS of the secondary for the relationship participating in the Global Mirror session. **CCA** CCA of the secondary for the relationship participating in the Global Mirror session.

Chapter 24. Managing Global Mirror

The parameters for managing a Global Mirror session can be modified simply by pausing and then resuming the session with the new parameters. The parameters that can be modified are:

- Consistency group interval time
- Maximum coordination interval time
- Maximum consistency group drain time

The topology of the Global Mirror session (master SSIDs, subordinate SSIDs, and subordinate serial numbers) cannot be changed without stopping and restarting the session. The session ID cannot be changed for an active session.

The RSESSION start or resume command may be issued to any device and CKD LSS in the master storage control. If you try to start or resume a device in a subordinate storage control, the request will fail. Once the session is active (initial configuration created), all START, PAUSE, RESUME and STOP commands to the master storage control must be issued to the same LSS where the start was issued as long as the configuration exists. IBM recommends that the same device in the master storage control be used for all START and RESUME commands.

In this topic

The following sections are included in this topic:

How Global Mirror consistency groups are formed

The frequency with which consistency groups are formed is controlled by the number of seconds you specify for the Consistency Group Interval in the RSESSION START or RESUME command, plus the number of seconds it takes to actually form the consistency group. This sum, shown in Figure 41 on page 440, is often referred to as the Recovery Point Objective (RPO). Point in time copies are written to the tertiary volume, or FlashCopy target, according to the timing of the RPO.

The user-specified consistency group interval is the interval between the completion or failure to form a consistency group and the start of an attempt to form the next consistency group. It typically takes 3-5 seconds to form a consistency group, so if 0 is specified for the consistency group interval, the RPO would be 3-5 seconds, not 0. However, a consistency group interval of 0 causes consistency groups to be formed continuously, starting a new consistency group as soon as the previous one is formed. A peak in write I/Os can increase the amount of time it takes to form a consistency group because of increased *drain time*, the time it takes to update the remote secondaries with the coordinated primary updates. An insufficient bandwidth (due to a link failure or insufficient link configuration) can also increase the amount of time it takes to form a consistency group.

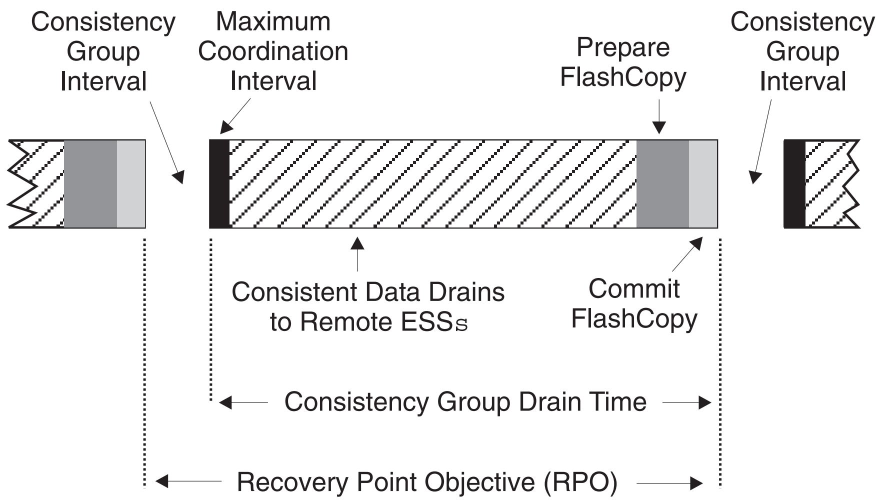


Figure 41. Forming Consistency Groups According to Specified Intervals

If a consistency group cannot be formed, data continues to flow to the secondary of the PPRC pair, but the point in time FlashCopy is not done.

A withdraw of a FlashCopy relationship within an Global Mirror session will impact the ability to form a consistency group. If you must withdraw a FlashCopy relationship, first remove the PPRC primary volume associated with the FlashCopy from the Global Mirror session with RVOLUME REMOVE, or pause the session with the RSESSION PAUSE command.

The suspension of a PPRC pair within a Global Mirror session will impact the ability to form a consistency group. If you must suspend a PPRC pair, first remove the PPRC primary volume from the Global Mirror session. If an error occurs causing a PPRC pair to be suspended, remove the pair from the session as soon as possible. Suspension of a PPRC pair within an active session will cause the next consistency group formation to fail.

As part of the consistency group formation, the master storage control must communicate with the subordinate storage controls to determine a consistent data point. A parameter called *Maximum Coordination Interval* (Figure 41) enables you to specify the maximum amount of time, in milliseconds, that the master storage control can spend communicating with storage control subordinates to determine a consistent data point. Since host write I/O is delayed while the consistent data point for a consistency group is being set, performance can be impacted when coordination cannot complete in an acceptable amount of time. If the Maximum Coordination Interval expires before formation of the consistent data point completes, the Consistency Group will be failed and the master will attempt to form the Consistency Group again after the next Consistency Group Interval time. The default value of the Maximum Coordination Interval is 50 milliseconds.

When a consistency group is formed, the data not yet copied must be transferred, or *drained*, to the PPRC secondaries at the remote site. During this time, new writes to the PPRC primaries are recorded in a change recording bitmap and not transferred to the PPRC secondaries. If one or more volumes in the consistency group have significantly more data to drain than the others, these volumes will prolong the time it takes to form the consistency group. This, in turn, can cause a backlog of updates on heavily updated primary volumes that must be transferred to the target volumes after the in-progress consistency group has been formed, resulting in an inefficient use of the PPRC links. A parameter called *Maximum Consistency Group Drain Time* (Figure 41 on page 440) enables you to reduce the impact on the primaries by causing the consistency group to fail when the amount of time expires and the data to be transferred to the PPRC secondaries has not completed. At this point, all volumes can continue to transfer data to the remote site. The default value of the Maximum Consistency Group Drain Time is 30 seconds.

In order to make sure that a Consistency Group does get formed eventually, the Global Mirror master keeps track of how many times the Maximum Consistency Group Drain Time has consecutively expired. If it has expired 5 times (causing the Consistency Group formation to fail five times), the timers will be disabled and the Consistency Group will be allowed to form, no matter how long it takes unless there is a communications failure or fatal error impeding the drain.

A pause will not interrupt the formation of a consistency group. If the formation of a Consistency Group is in progress when a pause command is received, the pause will take effect after the formation of the consistency group is complete. The RSESSION STOP command can interrupt the formation of a Consistency Group in progress, unlike RSESSION PAUSE, which allows Consistency Group formation to complete. RSESSION STOP can interrupt a consistency group formation in process as long there is a consistent set at the recovery site. RSESSION STOP with the FORCE parameter will stop the session even if consistent data cannot be formed.

Restrictions

The start or resume command may be issued to any device and CKD LSS in the master storage control. Once the Global Mirror session is active, all start or resume commands to the master storage control must be issued to the same CKD LSS where the start was issued, as long as the configuration exists. It is strongly recommended that the same device in the master storage control be used for all start and resume commands.

If the command to start or resume a session is issued without the consistency group interval, coordination interval, and coordination drain time specified, and the session is already active, the command is accepted.

Changing the Global Mirror session performance parameters

Once the Global Mirror session is started, you cannot issue an RSESSION START command to that session with different performance parameters specified for CGINTERVAL, CGDRAIN, or COORDINTERVAL. To change these parameters:

1. Issue the RSESSION PAUSE command to the same CKD LSS in the master storage control where the start was issued.
2. Issue the RSESSION RESUME command with the new performance parameters to any CKD device and LSS in the master storage control

If you issue RSESSION START with different performance parameters to a session that is either active or paused, the new parameters are ignored. If you issue RSESSION RESUME without specifying any performance parameters, the default values are applied and any custom values previously specified with START are lost. If you want to specify any performance parameter with RSESSION START or RSESSION RESUME, you must specify all three performance parameters.

Changing the Global Mirror session topology

If you wish to add storage control subordinates to an existing Global Mirror session, you must stop the session and start it again with the new ESS specified.

To add storage control subordinates to a session that is already started:

1. Issue the RSESSION STOP command to the same CKD LSS in the master storage control where the start was issued.
2. Use the PPRC Establish Path command to establish FCP paths between the master and any new subordinate storage controls that will be participating in the session.
3. Use the RSESSION DEFINE command to define the session to the LSSs in the new subordinate storage controls that are to participate in the session.
4. Issue the RSESSION START command to the master, specifying the updated topology and performance parameters for the session.

If additional LSSs are to be added to a session that is already started, and another LSS in the same ESS already belongs to the session (the ESS is already a master or subordinate), the session need not be stopped and restarted.

Adding volumes to the Global Mirror session

You can use the RVOLUME command to join volumes to the session anytime after the session ID is defined to the LSS where the volume resides. Once the session is started, you can add or remove volumes in the session, however, **the only volume that can be added is the primary of the PPRC pair**. Primary volumes may be added to a session in any state: simplex, pending, duplex, or suspended. If a volume joined to a session has not yet completed its initial copy, or if the proper FlashCopy relationship has not been established from the PPRC secondary to a target, its status in the session will be “join pending” and it will not participate in consistency group formation until the initial copy is completed. If a volume added to a session is the primary of a suspended pair, the first attempt to form a consistency group will fail, so this is strongly discouraged.

In a multi-target environment, only one of the pairs can participate in a Global Mirror session. The secondary needs to be identified at join time.

Terminating a Global Mirror session

To terminate a Global Mirror session and free all the resources in the session:

1. Issue the RSESSION STOP command to the same CKD LSS in the master storage control where the start was issued.
2. Issue the RVOLUME REMOVE command to remove the volumes from the session.
3. Issue the RSESSION UNDEFINE command to the same CKD LSS in the master storage control where the start was issued.

Failover/failback for a Global Mirror session

This topic describes the process of failing over from the primary site to the recovery site for planned and unplanned outages in a Global Mirror environment, as well as restoring the original recovery configuration at the primary site when it is back online. In these discussions, the volumes at the primary (home) site are referred to as the A volumes, and the volumes at the recovery site are referred to as the B volumes. FlashCopy volumes are referred to as C volumes.

Moving a Global Mirror session to the recovery site in a planned outage

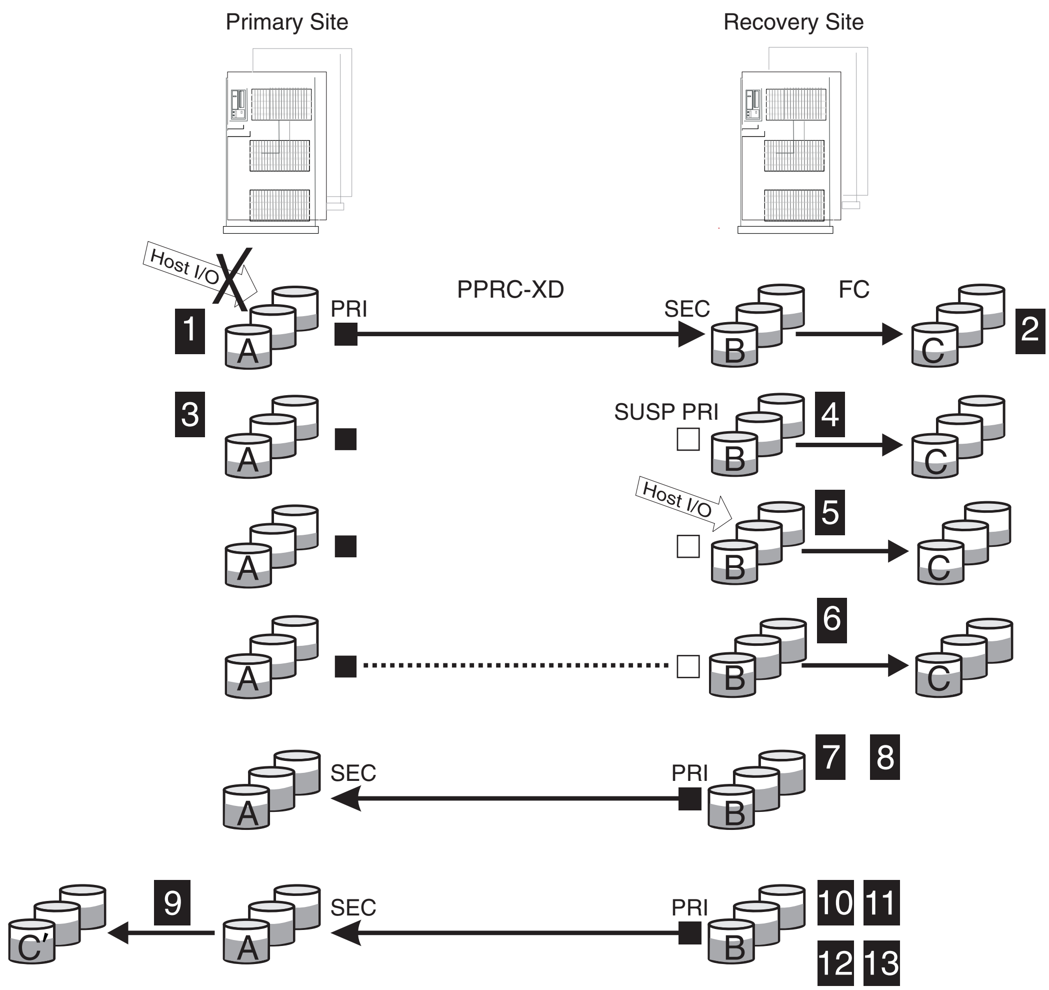


Figure 42. Moving a Global Mirror Session to Recovery Site in a Planned Outage

1. Quiesce applications at the primary site to reduce the risk of data loss.
2. Allow two consistency group formations to complete. Use the RQUERY command to confirm that the GoodCGCount increases by two.
3. Terminate the Global Mirror session using the RSESSION command specifying the STOP ACTION.
   a. If the master does not stop, reissue the command. If the master still does not stop, reissue RSESSION specifying the MASTER YES and FORCE parameters.

b. Issue RQUERY to determine if any storage control subordinates are still running.
c. Issue RSESSION to each subordinate storage control still running, specifying the MASTER NO parameter.

4. Issue the PPRC Establish Pair command to the B volume as the DEVN, specifying FAILOVER for ACTION, reversing the original primary and secondary parameters. This command changes the B volumes from secondaries to suspended primaries.
5. Issue the FCQUERY command to each of the suspended primary (B) volumes at the recovery site.

   Note: The revertible status (column headed RV) and the FlashCopy sequence numbers (column headed SEQNUM) in order to determine which step to perform next.
   - If all the FlashCopy relationships are non-revertible and the FlashCopy sequence numbers of the relations are equal, then the consistency groups on the C volumes are intact. Proceed to step 5.
   - If all the FlashCopy relationships are revertible and the FlashCopy sequence numbers of the relations are equal, then you must issue the FlashCopy Withdraw command specifying the REVERT ACTION to all FlashCopy relationships in the consistency group. This restores the relationships to their prior state, resets the revertible state, and removes write inhibit on the B volumes. When the FlashCopy Withdraw command completes, proceed to step 5.
   - If the FlashCopy sequence numbers of the relationships are equal and at least one of the relationships is in a non-revertible state, then you must issue a FlashCopy Withdraw specifying COMMIT to all of the FlashCopy relationships in the consistency group. This commits the FlashCopy relationships to the current state, resets the revertible state, and removes write inhibit on the B volumes. When the FlashCopy Withdraw command completes, proceed to step 5.
   - If more than two non-zero FlashCopy sequence numbers are present, then the consistency groups have been corrupted and the recovery program cannot continue.
6. Restart applications at the recovery site.

   Once a site switch is in progress, the Global Mirror session can now be reversed with the following steps.
7. Use the PPRC Establish Paths command to provide a path for data flow from the recovery site to the primary site. Paths are required between all LSSs in which PPRC pairs reside.
8. Issue the PPRC Establish Pair command to the B volume as the DEVN specifying the FAILBACK ACTION, again reversing the original primary and secondary parameters. This copies any changes from the B volumes to the A volumes.
9. Use the FlashCopy Withdraw command to end the FlashCopy relationship at the recovery site.
10. Start a new FlashCopy relationship between A and C′ volumes at the primary site, using the FlashCopy Establish command and specifying ASYNC with the MODE parameter.
11. Issue the RSESSION DEFINE command at the recovery site to specify the topology of the new Global Mirror session.

12. Use the PPRC Establish Paths command to provide a path for data flow between the Master and storage control subordinates at the recovery site.
13. Use the RVOLUME JOIN command to populate the session with primary volumes at the recovery site.
14. Issue RSESSION START to the Master at the recovery site to begin the Global Mirror session.

A Global Mirror session is now running at the recovery site, with the B volumes acting as the primary volumes, the A volumes acting as the secondary volumes, and the C′ volumes acting as the tertiary volumes.

Returning Global Mirror to the primary site in an planned outage

If a Global Mirror session is not running from the recovery site to the primary site, go to step 4.

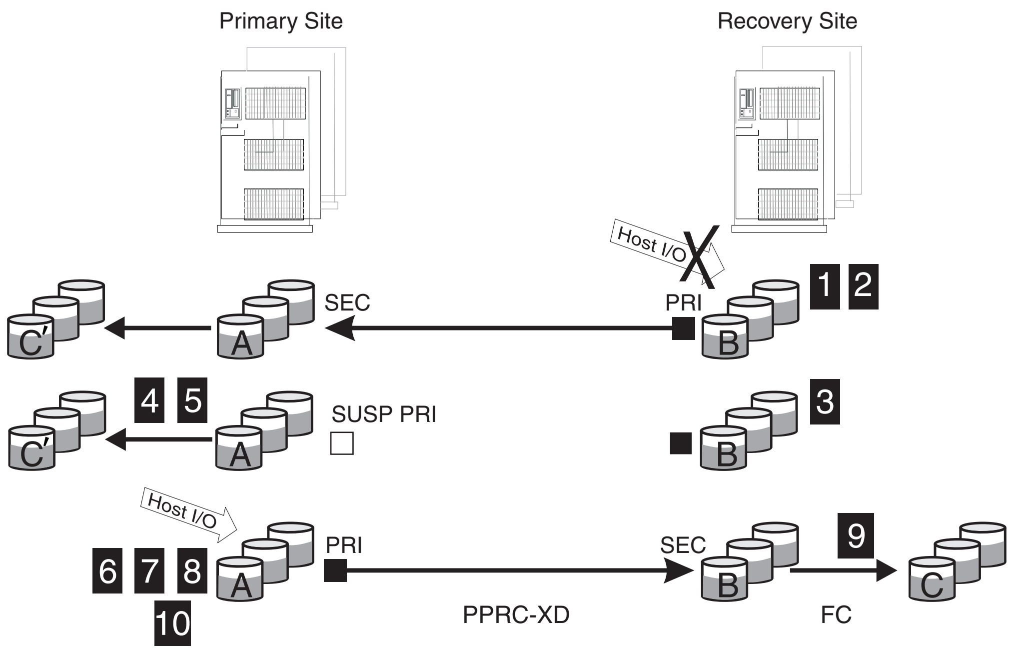


Figure 43. Returning Global Mirror to Primary Site in an Planned Outage

1. Quiesce applications at the recovery site to reduce the risk of data loss.
2. Allow two consistency group formations to complete. Use the RQUERY command to confirm.
3. Terminate the Global Mirror session using the RSESSION command specifying the STOP ACTION.
   a. If the master does not stop, reissue the command. If the master still does not stop, reissue RSESSION specifying the MASTER YES and FORCE parameters.
   b. Issue RQUERY to determine if any storage control subordinates are still running.
   c. Issue RSESSION to each subordinate storage control still running, specifying the MASTER NO parameter.
4. Verify that paths are still established from the primary site to the recovery site. Issue PPRC Establish Paths commands if necessary.

5. Issue the PPRC Establish Pair command to the A volume as the DEVN specifying the FAILOVER ACTION with the original PRIMARY and SECONDARY parameters. The A volumes now become suspended primaries with change recording active and write inhibit removed.
6. Restart applications at the primary site.
7. Issue the PPRC Establish Pair command to the A volume as the DEVN specifying the FAILBACK ACTION with the original PRIMARY and SECONDARY parameters to resynchronize the A and B volumes in the original recovery configuration.
8. If a Global Mirror session from the recovery site to the primary site was started, terminate the FlashCopy relationship at the primary site with the FlashCopy Withdraw command.
9. Start a new Flashcopy relationship between B and C volumes at the recovery site, using the FlashCopy Establish command and specifying ASYNC with the MODE parameter.
10. Issue RSESSION with the START ACTION to the Master at the primary site to begin the Global Mirror session. In a planned outage, the session would be stopped but remain defined in the LSS, and volumes would remain joined in the session. Thus, the session would not have to be redefined.

Moving a Global Mirror session to the recovery site in an unplanned outage

In step 4 on page 447 of the following scenario, column header names and data values are cited that appear only in formatted output from the RQUERY command. You may prefer unformatted DSECT output from RQUERY. The values from formatted output are used here only as examples.

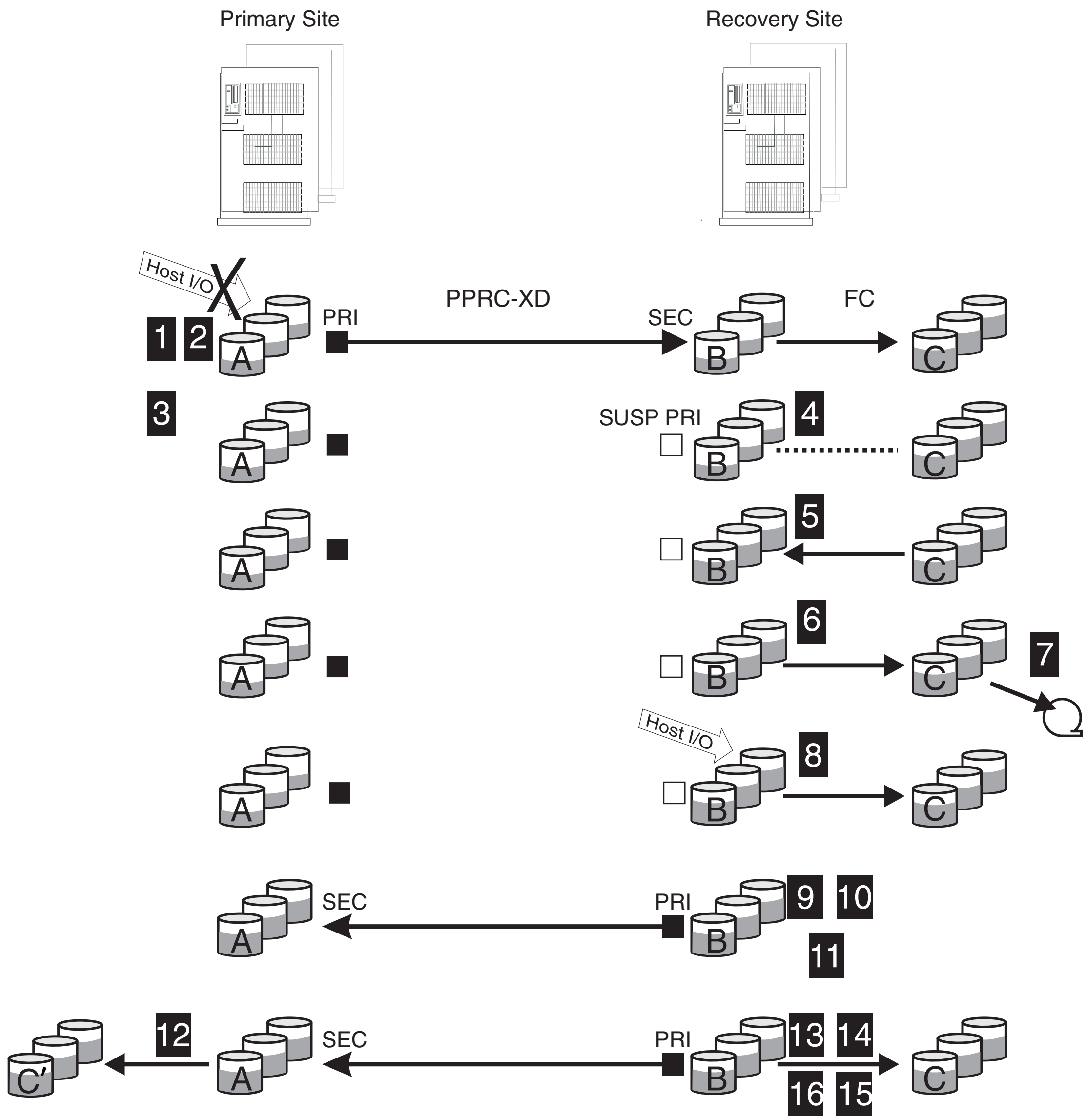


Figure 44. Moving Global Mirror Session to Recovery Site in an Unplanned Outage

1. Issue the RSESSION command with the STOP ACTION and MASTER YES parameters specified. If the master storage control is unable to complete consistency group formation, if possible reissue RSESSION STOP specifying MASTER YES and the FORCE parameter.
2. If the storage control subordinates have not stopped, issue the RSESSION STOP command, specifying NO for the MASTER parameter.
3. Issue the PPRC Establish Pair command to the B volume as the DEVN, specifying FAILOVER for ACTION and reversing the original PRIMARY and SECONDARY parameters. This establishes the B volumes as suspended primaries, activates change recording on the B volumes, and releases write inhibit on the B volumes.
4. Issue the RQUERY command to each of the suspended primary (B) volumes at the recovery site. Note the revertible status (column headed RvrtStat) and the FlashCopy sequence numbers (column headed FCSeqNbr) in order to determine which step to perform next.

- If all the FlashCopy relationships are non-revertible AND the FlashCopy sequence numbers of the relations are equal (RvrtStat = AllNRv&Eq), then the consistency groups on the C volumes are intact. Proceed to step 5.
- If all the FlashCopy relationships are revertible AND the FlashCopy sequence numbers of the relations are equal (RvrtStat = AllRv&Eq), then you must issue the FlashCopy WITHDRAW command specifying the REVERT ACTION to all FlashCopy relationships in the consistency group. This restores the relationships to their prior state, resets the revertible state, and removes write inhibit on the B volumes. When the FlashCopy WITHDRAW command completes, proceed to step 5.
- If the FlashCopy sequence numbers of the relationships are equal AND at least one of the relationships is in a non-revertible state (RvrtStat = Rv|NRv&Eq), then you must issue a FlashCopy WITHDRAW specifying COMMIT to all of the FlashCopy relationships in the consistency group. This commits the FlashCopy relationships to the current state, resets the revertible state, and removes write inhibit on the B volumes. When the FlashCopy WITHDRAW command completes, proceed to step 5.
- If more than two non-zero FlashCopy sequence numbers are present, then the consistency groups have been corrupted and the recovery program cannot continue.

5. Issue the FlashCopy Establish command specifying the C volumes as source and the B volumes as target, and specifying the FRR parameter. The B volumes are now consistent and the C volumes are no longer usable.
6. When the FlashCopy FRR command completes, issue another FlashCopy Establish command establishing the B volumes as source and the C as target in a new relationship to create a backup copy.
7. If desired, make a backup of the C volumes on tape for a safe copy.
8. Restart applications at the recovery site.

   If the primary site is operational, the Global Mirror session can be reversed with the following steps.
9. Use the PPRC Establish Paths command to provide a path for data flow from the recovery site to the primary site. Paths are required between all LSSs in which PPRC pairs reside.
10. Issue the CESTPAIR command specifying the B volume as the DEVN with the FAILBACK ACTION, again reversing the original PRIMARY and SECONDARY parameters. This copies any changes from the B volumes to the A volumes.
11. Use the FlashCopy Withdraw command to end the FlashCopy relationship at the recovery site.
12. Start a new FlashCopy relationship between A and C′ volumes at the primary site, using the FlashCopy Establish command and specifying ASYNC with the MODE parameter.
13. Issue the RSESSION DEFINE command at the recovery site to specify the topology of the new Global Mirror session.
14. Use the PPRC Establish Paths command to provide a path for data flow between the Master and storage control subordinates at the recovery site.
15. Issue the RVOLUME command specifying the JOIN ACTION to populate the session with primary volumes at the recovery site.
16. Issue the RSESSION command to the Master at the recovery site, specifying the START ACTION to begin the Global Mirror session.

Returning Global Mirror to the primary site in an unplanned outage

If a Global Mirror session is not running from the recovery site to the primary site, go to step 4.

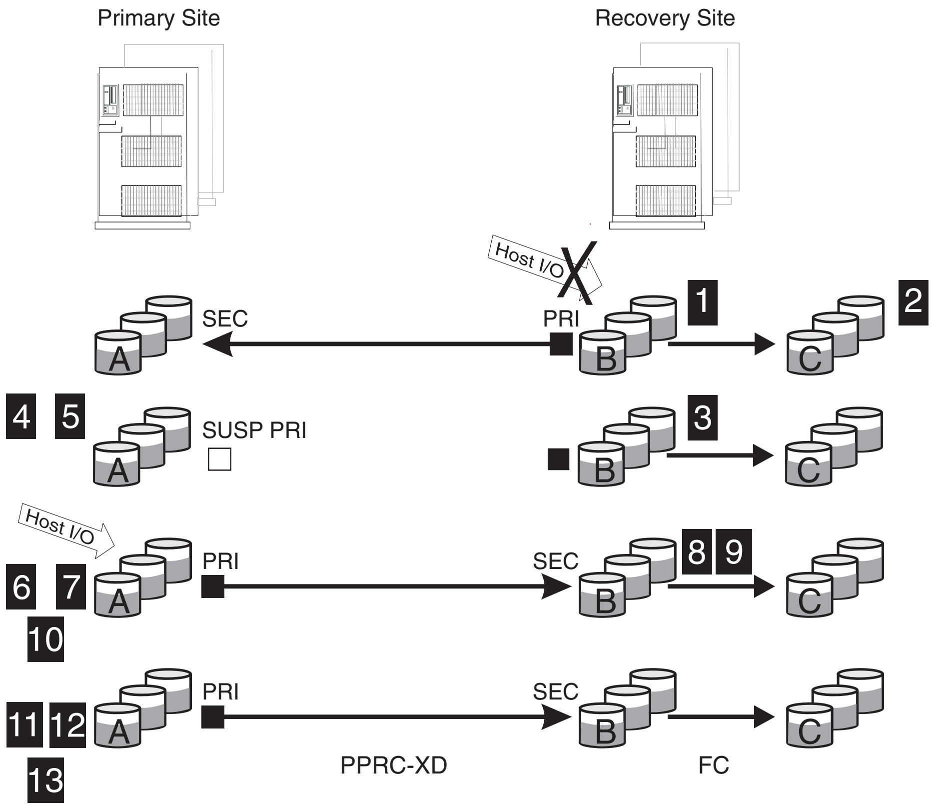


Figure 45. Returning Global Mirror to Primary Site in an Unplanned Outage

1. Quiesce applications at the recovery site to reduce the risk of data loss.
2. Allow two consistency group formations to complete. Use the RQUERY command to confirm.
3. Terminate the Global Mirror session using the RSESSION command specifying the STOP ACTION.
   a. If the master does not stop, reissue the command. If the master still does not stop, reissue RSESSION specifying the MASTER YES and FORCE parameters.
   b. Issue RQUERY to determine if any storage control subordinates are still running.
   c. Issue RSESSION to each subordinate storage control still running, specifying the MASTER NO parameter.
4. Verify that paths are still established from the primary site to the recovery site. Issue PPRC Establish Paths commands if necessary.
5. Issue the PPRC Establish Pair command to the A volume as the DEVN, specifying the FAILOVER ACTION with the original PRIMARY and SECONDARY parameters. The A volumes now become suspended primaries with change recording active and write inhibit removed.

6. Restart applications at the primary site.
7. Issue the PPRC Establish Pair command to the A volume as the DEVN, specifying the FAILBACK ACTION with the original PRIMARY and SECONDARY parameters to resynchronize the A and B volumes in the original recovery configuration.
8. If a Global Mirror session from the recovery site to the primary site was started, terminate the FlashCopy relationship at the primary site with the FlashCopy Withdraw command.
9. Start a new the Flashcopy relationship between B and C volumes at the recovery site, using the FlashCopy Establish command and specifying ASYNC with the MODE parameter.
10. If necessary, issue the RSESSION command at the primary site, with the DEFINE ACTION to specify the topology of the new Global Mirror session.
11. If necessary, issue the PPRC Establish Paths command at the primary site to provide a path for data flow between the Master and storage control subordinates at the recovery site.
12. If necessary, use the RVOLUME command at the primary site, specifying the JOIN ACTION to populate the session with primary volumes at the recovery site.
13. Issue RSESSION START to the Master at the primary site to begin the Global Mirror session.

Diagnosing Global Mirror session problems

You may find it useful to incorporate a polling method to monitor the Global Mirror session status to detect any problems that may be occurring. The following topics contain definitions for all Global Mirror status codes.

- "Querying Global Mirror summary output at session level" on page 430
- "Querying Global Mirror summary output at LSS level" on page 434
- "Querying Global Mirror summary output at device level" on page 436

See "Common end codes for REQUESTS within each ILK" on page 735 for other explanations and recovery actions.

Chapter 25. Metro/Global Mirror

In this topic

The following sections are included in this topic:

Introducing Metro/Global Mirror

The Metro/Global Mirror function enables a three-site, high availability disaster recovery solution. It combines the capabilities of both Metro Mirror and Global Mirror functions for greater protection against planned and unplanned outages. Metro/Global Mirror is supported across System z and open-systems environments. Metro/Global Mirror is supported on the DS8000®. It is not supported on the DS6000™ or the ESS800.

Metro/Global Mirror is a three-site solution, which uses synchronous replication to mirror data between a local site and an intermediate site, and asynchronous replication to mirror data from an intermediate site to a remote site. In this configuration, a Metro Mirror pair is established between two nearby sites (local and intermediate) to protect from local site disasters. The Global Mirror volumes can be located thousands of miles away and continue to be updated if the original local site has suffered a disaster and I/O has to be failed over to the intermediate site. In the case of a local-site-only disaster, Metro/Global Mirror can provide a zero-data-loss recovery at the remote site as well as at the intermediate site.

The Metro/Global Mirror function provides the following combination of synchronous and asynchronous mirroring:

- A nearby two-site synchronous copy that can protect from local disasters.
- A longer distance asynchronous copy, at a third site, that can protect from larger scale regional disasters. The third site provides an extra layer of data protection.

Metro/Global Mirror is an extension of Global Mirror, which is based on existing Global Copy (formerly known as PPRC XD) and FlashCopy functions. Global Mirror running at the intermediate site, using a master storage unit internally manages data consistency, removing the need for external software to form consistency groups at the remote site.

Figure 46 on page 453 shows the three sites that are used in a Metro/Global Mirror configuration. A Metro/Global Mirror environment is configured using a minimum of three storage units, one each at the local, intermediate, and remote sites. Data from the A volumes at the local site is synchronously replicated to B volumes at the intermediate site using Metro Mirror. Data from the B volumes at the intermediate site is asynchronously replicated to the C volumes at the remote site using Global Copy. FlashCopy relationships are created with the C volumes at the remote site as the FlashCopy source and the D volumes at the remote site as the FlashCopy target volumes, maintained as the consistent disaster recovery volumes using Global Mirror.

For Global Mirror, one storage unit at the intermediate site is designated as the master storage unit. The master storage unit sends commands over Fibre Channel Protocol (FCP) links and coordinates the consistency group formation process. These links are required for the Global Mirror master storage unit to coordinate the consistency group formation process with the storage units and to communicate the FlashCopy commands to the remote site. All status is relayed back to the master.

In a Metro/Global Mirror environment, if a PPRC pair between the local and intermediate sites is either duplex pending or becomes suspended, the Global Mirror session will fail consistency group formation. This is to ensure that the consistency group at the remote site is, in fact, truly consistent with the local site. There may be cases where the Metro Mirror secondary, at the intermediate site, is not aware that its status as a secondary is suspended, and consistency group formation would continue. The TSO RQUERY ACTION(DVCSTAT) or the API STAT4C query can provide information pertaining to the intermediate volume's status in the Global Mirror session and both of the cascaded PPRC volume's states (as a PPRC primary and as a PPRC secondary).

Setting up a Metro/Global Mirror configuration

Use this process to set up the environment to use Metro/Global Mirror.

To set up a Metro/Global Mirror Configuration, the remote site volumes do not have to be connected to the host site. The local and intermediate site volume needs to be connected to the site where the commands are going to be issued. If the intermediate site volumes are connected to an intermediate site system only, the Global Mirror setup commands are issued. If the local site has connectivity to the intermediate site and local site volume, then all of the commands can be issued from the local site.

Connect to the storage units to be used for System z or open systems host system storage.

Configure the following Metro/Global Mirror environment, which uses three sites (local, intermediate, and remote) and a minimum of four volumes (volume A, volume B, volume C, and volume D) on three storage units. For ease of description, the Metro/Global Mirror configuration is described in terms of A, B, C, and D volumes. Some environments can contain hundreds or thousands of volumes. See Figure 46 on page 453 for a high-level view of how the volumes are configured in a Metro/Global Mirror environment.

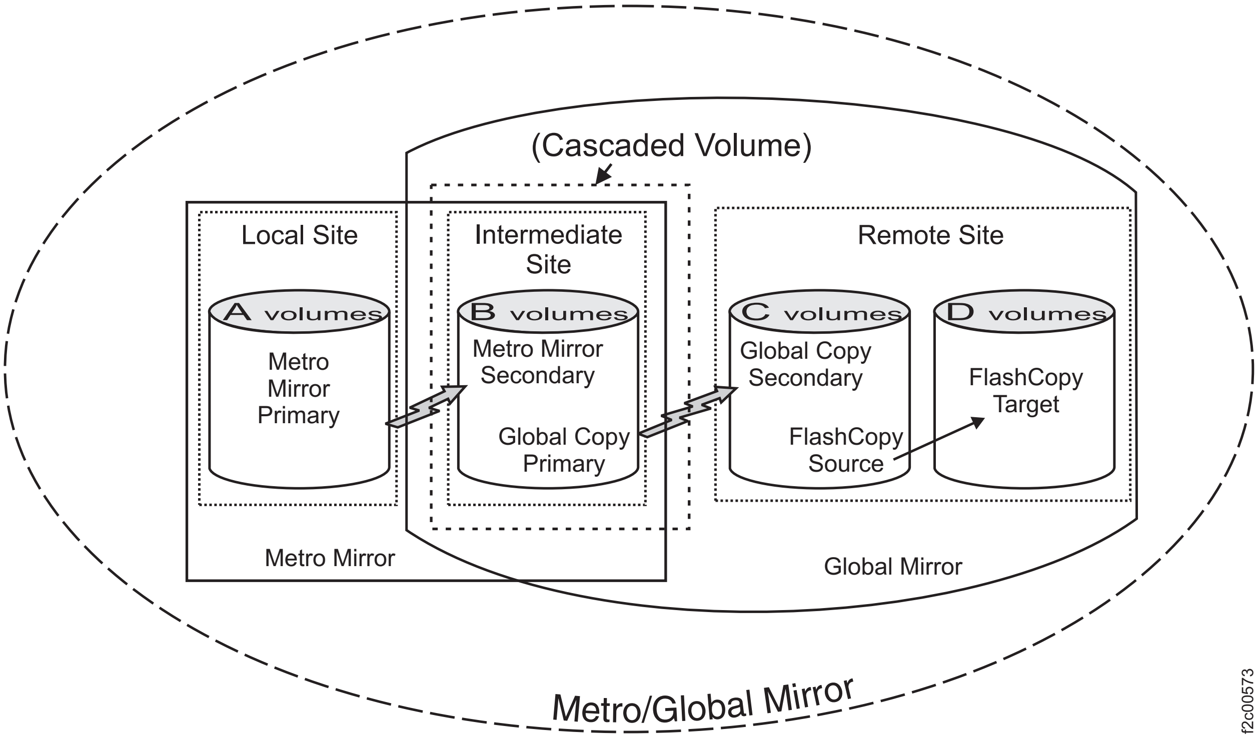


Figure 46. Metro/Global Mirror volume configuration

Before you set up a Metro/Global mirror configuration, do the following steps:

- Establish paths before you can establish pairs or start a session.
- Define a session to an LSS before you can add any volumes to the session.

To set up the environment to use Metro/Global Mirror, use the following process:

1. Establish paths between A volumes' LSSs and B volumes' LSSs.
2. Establish paths between B volumes' LSSs and C volumes' LSSs.
3. Establish paths from control unit that is the intended master of the Global Mirror session at the intermediate site and the intended subordinate control units at the intermediate site.
4. Define Global Mirror session to all LSSs that are going to participate in the session at the intermediate site.
5. Establish B volumes to C volumes Global Copy (PPRC-XD) relationships with the cascading and NOCOPY options.
6. Establish A volumes to B volumes PPRC relationships with the new option for change recording.
7. Establish C volumes to D volumes FlashCopy relationships, using ASYNC in the MODE keyword (for the Global Mirror remote site consistent copy) after the A volumes to B volumes relationships have reached a full duplex status.
8. Join the B volumes to the Global Mirror session.
9. Start the Global Mirror session.

Metro/Global Mirror recovery scenarios

For a Multi-Target Metro/Global Mirror configuration (where the primary has a Metro Mirror relationship with one PPRC pair and a Global Mirror relationship with the other) refer to *IBM DS8870 Version 7 Release 4 Multiple Target Peer-to-Peer Remote Copy* rather than using this topic.

In a Metro/Global Mirror configuration, if you lose access to the storage unit at the local site, the Global Copy relationship is intact and Global Mirror is still operational. Through the use of FlashCopy operations and consistency groups, consistent data is achieved at the remote site. However, you are left without the local site until all recovery operations have been performed to recover the local site.

A three-site Metro/Global Mirror configuration provides the following recovery options at the alternate sites if a failure occur:

- If an outage occurs at the local site, recovery operations can begin at the intermediate site. Global Mirror continues to mirror updates between the intermediate and remote sites, maintaining the recovery capability at the remote site.
- If an outage occurs at the local site, recovery operations can begin at the remote site and preparations can be made to resynchronize the local site when it recovers from its disaster. Once in recovery mode at the remote site, another Global Mirror session can be setup and put into operation using the former intermediate as its new remote site. This new Global Mirror session will provide additional disaster recovery solution while operating at the remote site.
- If an outage occurs at the intermediate site, data at the local storage unit is not affected. Applications continue to run normally.
- If an outage occurs at the remote site, data at the local and intermediate sites is not affected. Applications continue to run normally. The intermediate storage unit maintains a consistent up-to-date copy.
- If both the local and intermediate sites are lost, the scenario is similar to a two-site scenario when access to the local storage unit is lost. Recovery must be achieved using the last consistent point-in-time copy at the remote site.

Metro/Global Mirror planned site swap to intermediate

If a planned outage occurs, use the following process for failover and restore operations at the intermediate site:

1. At the local site, ensure that data consistency is achieved between the A and B volume pairs. This process will help coordinate the A volumes and B volumes consistency and allow consistent data to be copied to the remote site.

   You can use either one of the following methods to create data consistency:

   - Quiesce I/O processing to the A volumes at the local site.
   - Freeze write activity to the Metro Mirror primary volumes by performing the following steps:
     a. Freeze updates to the A volumes in Metro Mirror relationships across the affected LSSs. This ensures that the B volumes are consistent at the time of the freeze. (One command per storage unit or LSS is required.) As a result of the freeze action, the following actions are taken:
        – I/O processing to the Metro Mirror volume pairs is temporarily queued during the time when updates are frozen.

- The volume pairs that are associated with the source and target LSSs are suspended. During this time, the storage unit collects data that is sent to the A Metro Mirror volumes.
- The established paths between the logical subsystem (LSS) pairs are disabled.

b. Resume operations following a freeze. This operation is also called a thaw operation and it allows I/O processing to resume for the specified volume pairs.

2. Issue a PPRC establish pair command with the failover, force, and cascade options. Specify the remote site secondary devices (C volumes) as the primaries and the local site Metro Mirror primary devices (A volumes) as the secondaries.
3. Issue CRECOVER or PRECOVER commands to the intermediate devices (B volumes), for the Metro Mirror relationships (A volumes to B volumes).
4. Resume host I/O at the intermediate site.
5. Issue PPRC establish pair commands with the failback and force options from the remote site to the local site (site C to site A).
6. Wait for the first pass copy to complete from the remote to the local site (site C to site A), and then start the incremental re-synchronous change recording mechanism on the site B to site C pairs, with NOCOPY specified in the INCRESYNC parameter.
7. Use the RESSION command to stop Global Mirror at the intermediate site.
8. Suspend the Global Copy pairs between the intermediate and remote sites (site B to site C pairs).
9. Wait for all data to drain from the remote site back to the local site (site C to site A). An alternate from doing queries to wait for all out of sync data to drain is to convert the site C to site A pairs to synchronous. When they become full duplex, all data has been copied.
10. Suspend the site C to site A PPRC pairs.
11. Issue CRECOVER or PRECOVER commands to the remote devices (C volumes) to the B to C volumes, for the suspended site C to site B relationships.
12. Issue PPRC establish pair commands with failover and cascading to reverse the direction of the site C to site A relationships. The command is issued to the A devices.
13. Issue PPRC establish pair commands with failback to the site A to site C relationships.
14. Establish Metro Mirror by using the incremental resync change recording that has been in process since the site has swapped to the intermediate site, with the force option, from the intermediate to the local site (issue PPRC establish pair commands with option SYNC, mode INCRES, and force from site B to site A).
15. Issue PPRC establish pair commands to the site B to site A relationships to start the incremental resync change recording (issue PPRC establish pair commands with mode NOCOPY and INCRESYNC(START))
16. Wait for the site B to site A relationships to become full duplex and the site A to site C relationships to complete their first pass copy.
17. Start the Global Mirror, running from the local to the remote site in order to update the aged consistency group at the remote site.

The following steps show how to transit back to the original configuration (from site A to B to C):

1. Quiesce all host I/O activity and suspend the site B to site A relationships.
2. Issue a PPRC establish pair command with the failover, force, and cascade options. Specify the remote site devices (C volumes) as the primaries and the intermediate site devices (B volumes) as the secondaries.
3. Issue CRECOVER or PRECOVER commands to the local devices (A volumes), for the site B to site A relationships.
4. Resume host I/O at the local site.
5. Issue PPRC establish pair commands with the failback and force options from the remote site to the intermediate site (site C to site B).
6. Wait for the first pass copy to complete from the remote to the intermediate site (site C to site B), then start the incremental resynchronous change recording mechanism on the site A to site C pairs, with NOCOPY specified in the INCRESYNC parameter.
7. Use the RSESSION command to stop Global Mirror at the local site.
8. Suspend the Global Copy pairs between the local and remote sites (site A to site C pairs)
9. Wait for all data to drain from the remote site back to the local site (site C to site B). An alternate from doing queries to wait for all out of sync data to drain is to convert the site C to site B pairs to synchronous. When they become full duplex, all data has been copied.
10. Issue CRECOVER or PRECOVER commands to the remote devices (C volumes), for the suspended site C to site A relationships.
11. Suspend the site C to site B PPRC pairs.
12. Issue PPRC establish pair commands with failover and cascading to reverse the direction of the site C to site B relationships. The command is issued to the B devices.
13. Issue PPRC establish pair commands with failback to the site B to site C relationships.
14. Establish Metro Mirror by using the incremental resync change recording that has been in process since the site swap to the local site, with the force option, from the local site to the intermediate site (issue PPRC establish pair commands with option SYNC, mode INCRES, and force from A to B).
15. Issue PPRC establish pair commands to the site A to site B relationships to start the incremental resync change recording (issue PPRC establish pair commands with mode NOCOPY and INCRESYNC(START)).
16. Wait for the site A to site B relationships to become full duplex and the site B to site C relationships to complete their first pass copy.
17. Start the Global Mirror, running from the intermediate to the remote site in order to update the aged consistency group at the remote site.

You have changed the Metro/Global Mirror operations running from site A to B to C.

Metro/Global Mirror primary site unplanned outage – site swap to intermediate site

This scenario provides a disaster recovery solution if a failure occurs at your local site. You can run your operations from your intermediate site, which is protected by a two-site Global Mirror configuration, until your local site has recovered. Global Mirror continues sending updates to the storage unit at the remote site and continues to form consistency groups.

If an unplanned outage occurs, use the following process to perform failover and restore operations to the intermediate site.

1. At the local site, ensure that data consistency is achieved between the A and B volume pairs. Assume that the local site was not completely destroyed, it is essential that data from any surviving A and B volume pairs be copied and a consistent copy be achieved at the remote site. To create data consistency, freeze all write activity to the Metro Mirror primary volumes. If you quiesce host I/O to the A volumes at the local site, continue to step 2. If you freeze write activity to the Metro/Mirror source volumes, perform the following steps:
   a. Freeze updates to the A volumes in the Metro Mirror relationships across the affected LSSs. This ensures that the B volumes will be consistent at the time of the freeze process.
   b. Resume operations following a freeze. This operation is also called a thaw operation and it allows I/O processing to resume for the specified volume pairs.
2. Issue a PPRC establish pair command with the failover, force, and cascade options. Specify the Remote site secondary devices (C volumes) as the primaries and the local site Metro Mirror primary devices (A volumes) as the secondaries.
3. Issue CRECOVER or PRECOVER commands to the intermediate devices (B volumes), for the Metro Mirror relationships (A-B).
4. Resume host I/O at the intermediate site.

When the A volumes are ready to return to production, complete the following steps:

1. Issue PPRC establish pair commands with the failback and force options from the remote site to the local site (site C to site A).
2. Wait for the first pass copy to complete from the remote to the local site (site C to site A), and then start the incremental resync change recording mechanism on the site B to site C pairs, with NOCOPY specified in the INCRESYNC parameter.
3. Use the RSESSION command to stop Global Mirror at the intermediate site.
4. Suspend the Global Copy pairs between the intermediate and remote sites (site B to site C pairs).
5. Wait for all data to drain from the remote site back to the local site (site C to site A). An alternate from doing queries to wait for all out of sync data to drain is to convert the site C to site A pairs to synchronous. When they become full duplex, all data has been copied.
6. Suspend the site C to site A PPRC pairs.
7. Issue CRECOVER or PRECOVER commands to the remote devices (C volumes), for the suspended site C to site B relationships.
8. Issue PPRC establish pair commands with failover and cascading to reverse the direction of the site C to site A relationships. The command is issued to the A devices.
9. Issue PPRC establish pair commands with failback to the site A to site C relationships.
10. Establish Metro Mirror using the incremental resync change recording that has been in process since the site swap to the intermediate site, with the force

option, from the intermediate to the local site (issue PPRC establish pair commands with option SYNC, mode INCRES, and force from B to A).

11. Issue PPRC establish pair commands to the site B to site A relationships to start the incremental resync change recording (issue PPRC establish pair commands with mode NOCOPY and INCRESYNC(START)).
12. Wait for the site B to site A relationships to become full duplex and the site A to site C relationships to complete their first pass copy.
13. Start the Global Mirror, running from the local to the remote site in order to update the aged consistency group at the remote site.

Note: After you finish this process, the Metro/Global Mirror operations are running from site B to A to C. If you have your local and intermediate sites at the same location, the configuration is the same as the configuration from A to B to C. You can continue the operations.

To transition back to the original configuration (from site A to B to C), use the following process:

1. Quiesce all host I/O activity and suspend the site B to site A relationships.
2. Issue a PPRC establish pair command with the failover, force, and cascade options. Specify the remote site devices (C volumes) as the primaries and the intermediate site devices (B volumes) as the secondaries.
3. Issue CRECOVER or PRECOVER commands to the local devices (A volumes), for the site B to site A relationships
4. Resume host I/O at the local site.
5. Issue PPRC establish pair commands with the failback and force options from the remote site to the intermediate site (site C to site B).
6. Wait for the first pass copy to complete from the remote to the intermediate site (site C to site B), then start the incremental resync change recording mechanism on the site A to site C pairs, with NOCOPY specified in the INCRESYNC parameter.
7. Use the RSESSION command to stop Global Mirror at the local site.
8. Suspend the Global Copy pairs between the local and remote sites (site A to site C pairs).
9. Wait for all data to drain from the remote site back to the local site (site C to site B). An alternate from doing queries to wait for all out of sync data to drain is to convert the site C to site B pairs to synchronous. When they become full duplex, all data has been copied.
10. Issue CRECOVER or PRECOVER commands to the remote devices (C volumes) to the A to C volumes, for the suspended site C to site A relationships.
11. Suspend the site C to site B PPRC pairs.
12. Issue PPRC establish pair commands with failover and cascading to reverse the direction of the site C to site B relationships. The command is issued to the B devices.
13. Issue PPRC establish pair commands with failback to the site B to site C relationships.
14. Establish Metro Mirror by using the incremental resync change recording that has been in process since the site swap to the local site, with the force option, from the local site to the intermediate site (issue PPRC establish pair commands with option SYNC, mode INCRES, and force from site A to site B).

15. Issue PPRC establish pair commands to the site A to site B relationships to start the incremental resync change recording (issue PPRC establish pair commands with mode NOCOPY and INCRESYNC(START)).
16. Wait for the site A to site B relationships to become full duplex and the site B to site C relationships to complete their first pass copy.
17. Start the Global Mirror. Run from the intermediate to the remote site to update the aged consistency group at the remote site.

You have change the Metro/Global Mirror operations running from site A to B to C.

Metro/Global Mirror primary site planned outage – resume at remote

Use this process to perform failover and restore operations to your remote (C) site during a planned outage.

Before you issue a failover operation to the remote site, ensure that data processing has completely stopped at the local and intermediate sites. If you fail to do so and data is copied to the A and B volume pairs at the local and intermediate sites, a failover to the remote site can cause a data loss problem. This is the responsibility of the user, since it cannot be enforced by the TSO commands or the API.

This scenario describes the steps in which a failover operation is done to move production from the local site to a remote site and then a failback operation is done when processing is ready to return to the local site. Assume that host I/O cannot be sent to the local site in a Metro/Global Mirror configuration and it is not possible to run your systems using the B volumes at the intermediate site. You can switch operations to your remote site, which allows the processing of data to resume at the remote site. The Global Copy relationships between volumes at the intermediate and remote site are still operational. Global Mirror continues to operate between these two sites.

Follow these steps for failover and restore operations at the remote site:

1. At the local site, ensure that data consistency is achieved between the A and B volume pairs. You can use either one of the following methods to create data consistency:
   - Quiesce I/O processing to the A volumes at the local site.
   - Freeze write activity to the Metro Mirror primary volumes by performing the following steps:
     a. Freeze updates to the A volumes in Metro Mirror relationships across the affected LSSs. This ensures that the B volumes are consistent at the time of the freeze. (One command per storage unit or LSS is required.)
     b. Resume operations following a freeze. This operation also called a thaw operation and it allows I/O processing to resume for the specified volume pairs.
2. Verify that the last data from the local site has been included in a Global Mirror consistency group. Monitor this activity to determine when at least two consistency groups have formed since the local site I/O was quiesced or the freezes were issued. The total successful consistency group count field from the query output displays this information. At this point, the data on the B, C, and D volumes is consistent.
3. Stop the Global Mirror session.

4. Verify that the Global Mirror session has ended. Consistency groups will not be forming when Global Mirror processing is stopped.
5. Delete the relationships between the B and C volume pairs at the intermediate and remote sites. This prepares for reversing the direction of the volume pair from the remote site to the intermediate site. The cascaded relationship ends as well. Note: When the relationships between the B and C volumes are deleted, the cascade parameter is disabled for the B volumes and the B volumes are no longer detected as being in cascaded relationships.
6. Issue a failover command to the B and A volume pairs, with the Cascade option. With this process, updates are collected using the change recording feature, which allows for the resynchronization of the B and A volumes.
7. Create Global Copy relationships using the C and B volume pairs. Specify the NOCOPY option. Note: You can specify the NOCOPY option the B and C volumes contain exact copies of data.
8. Start I/O processing at the remote site. Continue in this mode until production is ready to return to the local site.
9. When you are ready to return production to the local site, quiesce I/O processing at the remote site. This process is used to begin the transition back host I/O to the A volumes.
10. Wait for the number of out-of-sync tracks on the C and B volume to reach zero. You can monitor this activity by querying the status of the C and B volumes. As soon as the number of out-of-sync tracks reaches zero, all data has been copied and the data on the C and B volumes is equal. All updates that are needed to resynchronize the A volumes are recorded at the B volumes.
11. Reestablish paths (that were disabled by the freeze operation) between the local site LSS and intermediate site LSS that contain the B to A Metro Mirror volume pairs.
12. Issue a failback command to the B volumes to A volume pairs. This command copies the changes back to the A volumes that were made to the B volumes while hosts were running on the B volumes. The A volumes are now synchronized with the B volumes.
13. Wait for the copy process of the B and A volume pairs to reach full duplex (all out-of-sync tracks have completed copying). You can monitor this activity by querying the status of the B and A volumes. As soon as the number of out-of-sync tracks reaches zero, all data has been copied and the data on the B and A volumes is equal. At this point, the data on volumes A, B, and C is equal.
14. Delete the Global Copy relationships between the C and B volume pairs between the intermediate and remote sites. Deleting the Global Copy relationships between the C to B volume pairs prepares for restoring to the original Global Copy relationships between the B to C volume pairs.
15. Issue a failover command to the A and B volume pairs. This process ends the Metro Mirror relationships between the B and A volumes and establishes the Metro Mirror relationships between the A and B volumes.
16. Reestablish paths (that were disabled by the freeze operation) between the local site LSS and the intermediate site LSS that contain the B to A Metro Mirror volume pairs.
17. Issue a failback command to the A volumes to B volumes. This command copies the changes back to the A volumes that were made to the B volumes in Metro Mirror relationships while hosts were running on the B volumes. The A volumes are now synchronized with the B volumes.

18. Reestablish the B to C volume pairs in Global Copy relationships. Specify the NOCOPY and the Cascade options.
19. Use FlashCopy to create a copy of C source volumes to the D target volumes, specifying the ASYNC option.
20. Restart Global Mirror processing.
21. Resume host I/O processing to the A volumes.

Metro/Global Mirror primary site unplanned outage – resume at remote

Use this process to perform failover and restore operations at your remote site during an unplanned outage, using E volumes at the intermediate site.

If possible, before you issue a failover operation to the remote site, ensure that data processing has completely stopped at the local and intermediate sites and issue a freeze to the A volumes at the local site. If you fail to do so and data is copied to the A and B volume pairs at the local and intermediate sites, a failover to the remote site can cause a data loss.

For this scenario, assume that host I/O processing is being sent to the local site in a Metro/Global Mirror configuration. A failure occurs at Site A and it is not possible to run your systems using the B volumes at the intermediate site. You can switch operations to your remote site (Site C), which allows the processing of data to resume at Site C. This process is known as a failover recovery. The Global Copy relationship between volumes at the intermediate and the remote sites are still operational. Global Mirror continues to operate between these two sites.

Perform the following steps after a failure has been detected at the local site:

1. Verify that the last data from the local site has been included in a Global Mirror consistency group. Monitor this activity by querying the B and C volumes to determine when at least two successful consistency groups have formed. The **Total Successful CG Count** field from the query output displays this information.
2. Stop the Global Mirror session from which the B and C volume pairs are included.
3. Verify that the Global Mirror session has ended. Consistency groups will not be forming when Global Mirror processing is stopped.
4. Delete the Global Copy relationships between the B and C volume pairs at the intermediate and remote sites. This prepares for reversing the direction of the volume pair from the remote site to the intermediate site. The cascaded relationship ends as well.

   Note: When the relationships between the B and C volumes are deleted, the cascade parameter is disabled for the B volumes and the B volumes are no longer detected as being in cascaded relationships.
5. Issue a failover command to the B volumes with the Cascade option. With this process, updates are collected using the change recording feature, which allows the resynchronization of the B to A volumes.
6. Create Global Copy relationships using the C and B volume pairs. Specify the NOCOPY option.

   Note: You can specify the NOCOPY option because the B and C volume pairs contain exact copies of data.

7. Use FlashCopy to create a copy of B source volumes to E target volumes. Specify the MODE(ASYNC) options. This creates a backup copy of the consistency group.
8. Create a Global Mirror session using the C volumes:
   a. Establish paths between the intended master and the intended subordinates at the remote site.
   b. Define the Global Mirror session to all of the LSSs that are going to participate in the session at the remote site.
   c. Join the C volumes to the Global Mirror session.
9. Start the Global Mirror session from which the C, B and E volumes are included.
10. Verify that the Global Mirror session has started.
11. Allow the I/O to run and monitor the formation of the consistency groups.
12. When the local site is ready to return, issue a failback command to the B and A volumes. This command copies the changes back to the A volumes that were made to the B volumes while hosts were running on the B volumes. The A volumes are now synchronized with the B volumes.
13. Wait for the copy operation of the B and A volumes to reach full duplex status (all out-of-sync tracks have completed copying). You can monitor this activity by querying the status of the B and A volume pairs.
14. End I/O processing to the C volumes.
15. Verify that at least two consistency groups have formed. Assume that consistency groups formed successfully, data in the A, B, C, and E volumes is consistent.
16. Stop the Global Mirror session between the C, B, and E volumes.
17. Verify that the Global Mirror session for which the C, B, and E volumes are included, has stopped.
18. At the remote site, remove the C volumes (or Global Copy secondary volumes) from the Global Mirror session that includes the C, B, and E volumes.
19. Delete the Global Copy relationships between the C to B volumes between the intermediate and remote sites. Delete the Global Copy relationships between the C to B volume pairs prepares for restoring to the original Global Copy relationships between the B to C volume pairs, which is described in step 23. The cascaded relationship ends, as well.
20. Issue a failover command to the A volumes. This process ends the Metro Mirror relationships between the B and A volumes and establishes the Metro Mirror relationships between the A and B volume pairs.
21. Reestablish paths (that were disabled by the freeze operation) between the local site LSS and intermediate site LSS that contain the B to A Metro Mirror volume pairs.
22. Issue a failback command to the A and B volumes. This command copies the changes back to the A volumes that were made to the B volumes in Metro Mirror relationships while hosts were running on the B volumes. The A volumes are now synchronized with the B volumes.
23. Establish the B and C volume pairs in Global Copy relationships. Specify the NOCOPY and Cascade options.
24. Optionally, you can issue a FlashCopy operation to create a backup copy of all the C, B, and E volumes from which the last consistency group was created. If you need to preserve data from the set of volumes (or consistency group) at was created using the E volumes, allow the background copy from the

FlashCopy process to complete before you continue to the next step, which describes removing the FlashCopy relationship between the B to E volume pairs.

25. Delete the FlashCopy relationship between the B and E volume pairs to end the relationship at the intermediate site.
26. Resume Global Mirror at the intermediate site. This starts Global Mirror processing for the B, C , and D volumes.
27. Resume I/O on A volumes.
28. Verify that consistency groups are forming successfully.

Incremental Resync for Metro/Global Mirror cascade

This topic describes Incremental Resync for Metro/Global Mirror for cascaded volumes.

For information on converting a cascaded configuration to a multi-target configuration, refer to "Converting to Multi-Target Mirror" on page 275.

Introducing Incremental Resync for Metro/Global Mirror

Incremental Resync for Metro/Global Mirror is used in a Metro/Global Mirror Peer-to-Peer Remote Copy configuration in order to maintain a backup site if one of the three sites is lost. The purpose of the Incremental Resync function is to avoid always having to do a full volume resynchronization between the local and remote sites if an outage occurs at your intermediate site. When the intermediate site is lost, the local and remote sites can be connected, copying a subset of the data on the volumes to maintain a backup site.

The microcode records the bytes in flight at the local site. If the intermediate site is lost, then the local site volume may be established as the primary to the volume at the remote site.

Incremental Resync uses a change recording mechanism to track in flight changes. Because there can only be one function that uses change recording active on a device at a time, incremental resync prevents change recording for other purposes, such as incremental FlashCopy. If an attempt is made to start Incremental Resync when another function using change recording is active, that start request fails. If an attempt is made to do an incremental FlashCopy from a PPRC primary that has Incremental Resync active, that FlashCopy request fails.

Like XRC, two change recording bitmaps are maintained. When data in flight is secured at the remote sight, the system updates the bitmaps. The local volume updates the change recording bitmaps checking the status of consistency group formation.

Setting up incremental resync for Metro/Global Mirror

The Incremental Resync for Metro/Global Mirror change recording mechanism can be started in the following conditions:

- When a Peer-to-Peer Remote Copy (PPRC) pair is established.
- A subsequent Establish PPRC command (by TSO or API) with the appropriate keyword is issued on an already established pair.

  To start the Incremental Resync, specify the INCRESYNC(START) when you use the CESTPAIR command to specify PPRC primary and secondary volumes. See "CESTPAIR – establishing volume pairs" on page 296 for details.

The incremental Resync for Metro/Global Mirror stops changing recording in the following conditions:

- The establish PPRC command with INCRESYNC(STOP) is issued.
- The PPRC pair is deleted.

Note:

1. If the establish PPRC command is issued and results in more than two secondaries for the specified PPRC primary, the command is rejected.
2. If an establish PPRC command with the start incremental resync keyword is issued to an already established PPRC pair that has the incremental resync change recording mechanism already started, the command is accepted and no action is taken.
3. If an establish PPRC command with the stop incremental resync keyword is issued to an already established PPRC pair that does not have the incremental resync change recording mechanism already started, the command is accepted and no action is taken.

If the intermediate site is lost, you can continue your operations by issuing an Establish PPRC command with Incremental Resync mode specified to connect the local volume to the remote volume. To retain consistency over distance, the pairs are established as Global Copy (extended distance, or MODE(XD)) and the Global Mirror environment created at the local site (define session to LSS'es involved, add volumes to session, and start session at the local site).

Note:

1. If this Establish PPRC command for recovery is issued and the Incremental Resync change recording mechanism does not run, the command is rejected.
2. If this Establish PPRC command for recovery is issued and specified primary volume is not a primary volume whose secondary is the primary for the specified secondary volume, and FORCE is not specified, the command is to be rejected.

With the incremental resync support installed, if there is an intermediate site failure, the procedure to restart the recovery environment running Global Mirror from the local site to the remote site is simplified, and a full copy is not required.

Recovery scenarios for Metro/Global Mirror with incremental resync

In a Metro/Global Mirror configuration, if you lose access to the storage unit at the intermediate site, the recovery scenario involves restarting the 2-site Global Mirror environment between the local and remote sites. In order to achieve this without having to do a full copy of the volumes from the local site to the remote site, the new function, Incremental Resync, is used.

Metro/Global Mirror planned intermediate site outage

With the incremental resync support installed, if there is an intermediate planned outage, to restart the recovery environment running Global Mirror from the local site to the remote site, do the following steps:

Assumptions:

1. Quiesce of host applications is not needed because operations will continue to run from the local site.

2. Metro Mirror pairs between the local and intermediate sites have been suspended in preparation for the planned outage.
3. Global Mirror session at the intermediate site has been stopped in preparation for the planned outage.

Recovery steps:
1. Perform failover commands to the C volumes at the remote site (PPRC establish C volumes to B volumes) with the cascade option.
2. Establish paths between A volumes' LSSs and 'C' volumes' LSSs.
3. Establish A to C pairs using the incremental resync change recording bitmaps (specify INCRES in the MODE parameter) to copy changed tracks. The configuration at this point is from volumes A to C, and then C to B, with A to C running PPRC-XD and the C to B pair being in failover state.
4. Establish paths from control unit that is the intended master of the global Mirror session at the primary (local) site and the intended subordinate control units at the primary site.
5. Define the Global Mirror session to all LSSs that are going to participate in the session at the primary site.
6. Join the A volumes to the Global Mirror session.
7. Start the Global Mirror session at the local site.

Note: To decrease the amount of time needed, steps 2, 4, 5, and 6 can be done before an outage, but the session is not started until needed.

After the intermediate site has been recovered, to restore operations to the original Metro/global Mirror environment, do the following steps:
1. Establish paths from the remote site C LSSs to the intermediate site B LSSs, if necessary.
2. At the intermediate site secondary devices (B volumes), issue CRECOVER or PRECOVER commands to terminate the relationships from A to B.
3. Suspend the relationships from B to C, if the relationships have not been suspended.
4. Perform Failback from C volumes to B volumes with the cascade option.
5. Start Incremental Resync on the A to C volume pairs with the INCRESYNC NOCOPY option.

   Note: This step is used to enable the A to B pairs to be restored by using the Incremental Resync change recording and the FORCE option.
6. Stop the Global Mirror session between the local and remote sites. Allow the data to continue to cascade to the C and B volumes using Global Copy.
7. Suspend the A to C Global Copy pairs, and wait for all data to drain from the C to the B volumes.
8. At the remote site Global Copy secondary devices (C volumes), issue CRECOVER or PRECOVER commands to terminate the relationship from A to C.
9. Suspend the relationship from C to B.
10. Issue established PPRC commands with failover from B to C and with the cascade option.

Note: At this point, A is a primary to C and B is in failover state to C, but C is still a cascaded state from A to B, and is suspend, because no data is being transferred from the local site (A volumes).

11. Perform the failback from B to C by issuing the Failback commands at site B with the cascade option
12. Establish A to B pairs using the incremental resync change recording bitmaps (INCRES in the MODE option) to copy changed tracks from A to B with **YES** specified for the FORCE option to bypass relationship verification.
13. Start the Incremental Resync function for the local PPRC volume pairs (A volumes to B volumes).
14. After the A volumes to B volumes pairs have reached full duplex, start the Global Mirror session at the intermediate site.

Metro/Global Mirror unplanned intermediate site outage

With the incremental resync support installed, if there is an intermediate site failure, to restart the recovery environment running Global Mirror from the local site to the remote site, do the following steps:

Assumptions:

1. Metro Mirror pairs between the local and intermediate sites will suspend if host writes are done to the local, or A volumes. If not, A to B pairs need to be suspended.
2. Global Mirror is one of the following status:
   - Global Mirror is no longer running.
   - Global Mirror has gone FATAL.
   - Global Mirror is no longer forming consistency groups.
3. Depending on the type of failure at the intermediate site, assume the worst scenario: Global Mirror was in the middle of forming a consistency group or performing the FlashCopy step as part of a consistency group, and was unable to complete. The consistency group will need to be verified at some point.

Recover steps:

1. Perform failover commands to the C volumes at the remote site (PPRC establish C volumes to B volumes) with the cascade option.

   Note: If you want to stop the Global Mirror session, stop the session after this step. If a stop issued to the master fails, a intermediate site failure might have happened in the middle of Global Copy formation. Thus, the Global Mirror and FlashCopies need to be reverted or committed. In this scenario, you can use the MASTER=NO parameter to stop the Global Mirror session on any subordinates that might have been orphaned during the outage.
2. Establish paths between A volumes' LSSs and C volumes' LSSs.
3. Establish A to C pairs using the incremental resync change recording bitmaps to copy changed tracks. The configuration at this point will be from volumes A to C, and then C to B, with A to C running PPRC-XD and the C to B pair being in failover state.
4. Establish paths from control unit that is the intended master of the global Mirror session at the primary (local) site and the intended subordinate control units at the primary site.
5. Define the Global Mirror session to all LSSs that are going to participate in the session at the primary site.
6. Join the A volumes to the Global Mirror session.

7. Start the Global Mirror session at the local site.

When the intermediate site has been recovered, to return to the original Metro/Global Mirror configuration, the recovery procedure is the same as that for recovery after a planned outage.

Part 5. FlashCopy

This provides the information to help you use the FlashCopy copy function.

Chapter 26. What is FlashCopy?

This section explains what the FlashCopy copy function is and how it is invoked using DFSMSdss. It also details the commands that are available for using the FlashCopy capabilities of the IBM TotalStorage Enterprise Storage Server (ESS) storage subsystem.

In this topic

The following sections are included in this topic:

Overview of FlashCopy

FlashCopy enables you to make copies of a set of tracks, with the copies immediately available for read or write access. This set of tracks can consist of an entire volume, a data set, or just a selected set of tracks.

FlashCopy provides both source volume to target volumes support, which came with FlashCopy Version 1, and source data set level to target data set level support, which comes with FlashCopy Version 2. FlashCopy can be used in combination with XRC, synchronous PPRC, and PPRC-XD.

The primary objective of FlashCopy is to create a copy of a source volume on the target volume. This copy is called a point-in-time copy. Access to the point-in-time copy of the data on the source volume is through reading the data from the target volume. The actual point-in-time data that is read from the target volume might or might not be physically stored on the target volume. As soon as a FlashCopy relationship is established (more specifically, as soon as the initialization process for a FlashCopy establish, initiated with the FCESTABL command, is complete), the point-in-time data is available for reading from the target volume. However, if data is written to a track that is a target track in a FlashCopy relationship and the updated target track is read afterwards, the data that is returned is user-updated data, and not the point-in-time source track data. Target tracks are withdrawn from a FlashCopy relationship as soon as any application writes to these tracks.

FlashCopy V1 requires the entire source volume and target volume to be involved in a FlashCopy relationship, even if selected tracks were specified on the FCESTABL command. FlashCopy V1 relationships do not allow any other FlashCopy relationships to exist on either the source or target volume.

ESS FlashCopy Version 2 enhances the FlashCopy function by providing an alternative method to copying an entire source volume to a target volume. This enhancement includes the following features:

- Multiple FlashCopy relationships are allowed on a volume.
- Track relocation is possible because for tracks to be copied, the target tracks do not need to be in the same location on the target volume as on the source volume.
- A FlashCopy target and source volume need not be in the same logical subsystem (LSS) in an ESS. However, FlashCopy must be processed in the same ESS.
- Extent level (data set level) FlashCopy.
- Incremental/Persistent Flashcopy.
- NOCOPY to COPY conversion.

Guidelines: Do not use the TSO or the API FlashCopy functions to copy data sets that you intend to access from the target volume. With FlashCopy, TSO or API usage does not provide any data management services, such as allocation or cataloging. Data sets that are copied using these functions are not accessible from the target volume without the user manually performing these data management tasks. You can use a data set copy program that provides these data management services as part of the copy process, such as DFSMSdss.

Understanding how FlashCopy works

When a FCESTABL command is processed, an attempt is made to create a relationship between tracks on a source device and tracks on a target device. A track extent contains a beginning track, an ending track, and all the tracks between the beginning track and the ending track. For FlashCopy V2, a source track extent and a target track extent are required to describe a track set. A contiguous set of source tracks related to a contiguous set of target tracks is called a track set. Each track set makes up a FlashCopy relationship.

You can request a FlashCopy relationship using:

- TSO/E commands
- An application programming interface (API) macro (ANTRQST) using the REQUEST=FCESTABLISH command. See Appendix C, "ANTRQST and ANTRQSTL macros – call to the system data mover API," on page 573.
- A REXX exec that calls program ANTTREXX. ANTTREXX uses the ANTRQST API. See Appendix D, "REXX support for the ANTRQST API," on page 745.
- A Web-browser interface called "IBM TotalStorage Enterprise Storage Server Copy Services." This Web-enabled interface is part of the ESS storage subsystem.
- DFSMSdss
- ICKDSF

A FlashCopy relationship can be established in:

- COPY mode, which runs a background copy process. This is the default.
- NOCOPY mode, which suppresses the background copy.

A FlashCopy relationship begins when the FlashCopy relationship is initiated and ends when the background copy completes or when you withdraw it, which you can do with the FCWITHDR command. When you use the FCESTABL command with the NOCOPY parameter to establish the FlashCopy relationship, you must explicitly withdraw the FlashCopy relationship when it is no longer required. You can specify one or more track sets using the FCWITHDR command, but each track set must have a source extent and a target extent.

You are not informed when the background copy is complete. To monitor when the copy completes, issue the FCQUERY command. A non-persistent FlashCopy relationship is terminated and this inactivity is reflected in the FCQUERY output. However, an incremental relationship, which is persistent, continues to appear as active in the FlashCopy report.

How long the actual physical copy takes depends on:
- The amount of data being copied
- The number of background copy processes that are occurring
- The other activities on the ESS.

When the FCESTABL command includes the MODE(COPY) parameter, the ESS copies all specified source tracks to specified target tracks. Figure 47 illustrates a full-volume copy, where the source tracks are copied to the target volume in the same track locations as the source tracks.

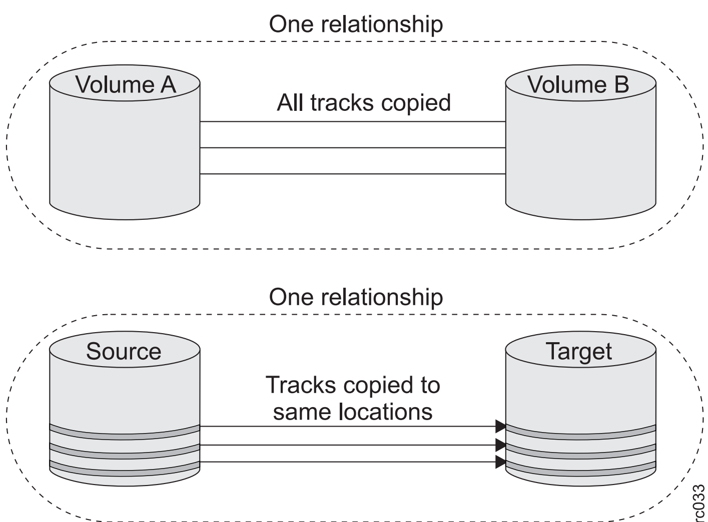


Figure 47. A full-volume copy, only one FlashCopy relationship active on a volume

You can copy the same source volume track to different tracks on one or more target volumes or even copy to a track on the source volume as a target, up to twelve times. Figure 48 on page 474 illustrates this concept.

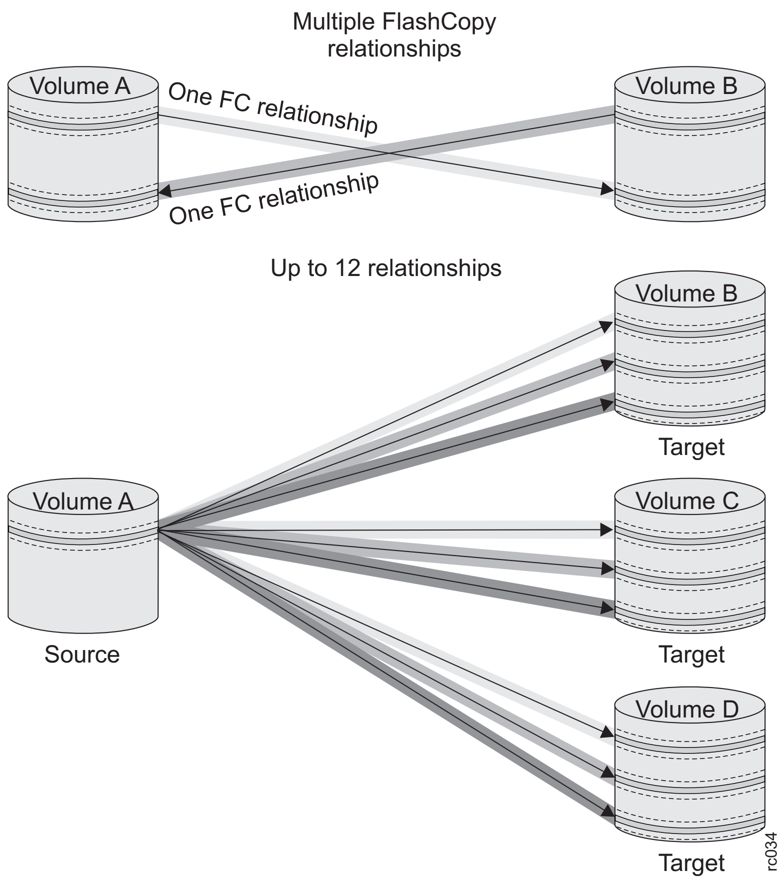


Figure 48. Multiple FlashCopy relationships active on a volume

Note: As long as a track on a volume is not a source or target track of an existing FlashCopy relationship, it can become the target track in a new FlashCopy relationship. The only restriction is that you cannot exceed the total number of active FlashCopy relationships per volume. Use the FCQUERY command to determine what that number is.

There might be circumstances that require you to withdraw a FlashCopy relationship. For example, you might not want to wait for the copy to complete or there might no longer be a need for the FlashCopy relationship. For either case, you can use the FCWITHDR command.

For additional information about:

- Configuring copy services options of the ESS, refer to *IBM TotalStorage Enterprise Storage Server Web Users Interface Guide*.
- ANTRQST API, refer to Appendix C, "ANTRQST and ANTRQSTL macros – call to the system data mover API," on page 573.
- ANTTREXX program, refer to Appendix D, "REXX support for the ANTRQST API," on page 745.
- FCESTABL command , refer to the "FlashCopy establish (FCESTABL) command" on page 499 and "Background copy of FlashCopy tracks" on page 475.
- FCQUERY command, refer to "FlashCopy query (FCQUERY) command" on page 510.
- FCWITHDR command, refer to "Withdrawing FlashCopy relationships" on page 489.

- ICKDSF, refer to*Device Support Facilities (ICKDSF) User's Guide and Reference*.

Background copy of FlashCopy tracks

Use the FCESTABL command to establish a FlashCopy relationship between the user-specified source and target devices using one of the following two modes:

- **COPY** runs a background copy. This is the default operation.

  The COPY mode causes all tracks in the FlashCopy relationship to be physically copied from the source volume tracks to the target volume tracks. For FlashCopy V1 operations, the source tracks are copied to the target volume in the same track locations as the source tracks. For FlashCopy V2 operations, source tracks are copied to the target volume in the track locations specified in the FlashCopy establish request.

  As soon as the relationship is established, user programs have access to two logical copies of the source data. Updates to the source volume after the FlashCopy relationship is established are *not* part of the logical copy on the target device.

  Data that is read from the target device returns source track data from the point-in-time of the FlashCopy establish. This is done without waiting for the physical track copy to complete. The FlashCopy relationship ends for each track set when it is copied to the target device. If you update a track on a target device while source tracks are being copied, the data on the target track contains your updated data. The updated track is withdrawn from the FlashCopy relationship. A read operation from the target device for tracks that you updated always returns your updated data.

  After the background copy is complete, the data on the target device is the same as the source device when the FlashCopy relationship was first established. **Exception:** If you modify the target tracks within the track extents, the data on the target device is not the same as the source device.

  When the background copy operation completes, the FlashCopy relationship ends and the ESS resources that maintain the relationships are released.

  Note: Background copy is not allowed with space efficient targets.

- **NOCOPY** suppresses the background copy.

  The FlashCopy NOCOPY mode causes the relationship to be established without immediately initiating a background copy. When the ESS receives an update to a source track in a FlashCopy relationship, a copy of the point-in-time (pre-update) data is preserved on the target volume. Data that is read from the target device returns source track data from the point-in-time of the FlashCopy establish.

  A FlashCopy relationship established in NOCOPY mode remains active until one of the following occurs:

  – All specified source device tracks are updated by user applications.

  – The ESS copies all tracks from source to target when a threshold number of source tracks are updated.

  – A FlashCopy withdraw (using the FCWITHDR command) is issued to remove the FlashCopy relationship.

  – A NOCOPY relationship is converted to a COPY relationship through the NOCOPY2COPY function.

  Figure 49 on page 476 shows how the NOCOPY relationship affects the source and target volumes.

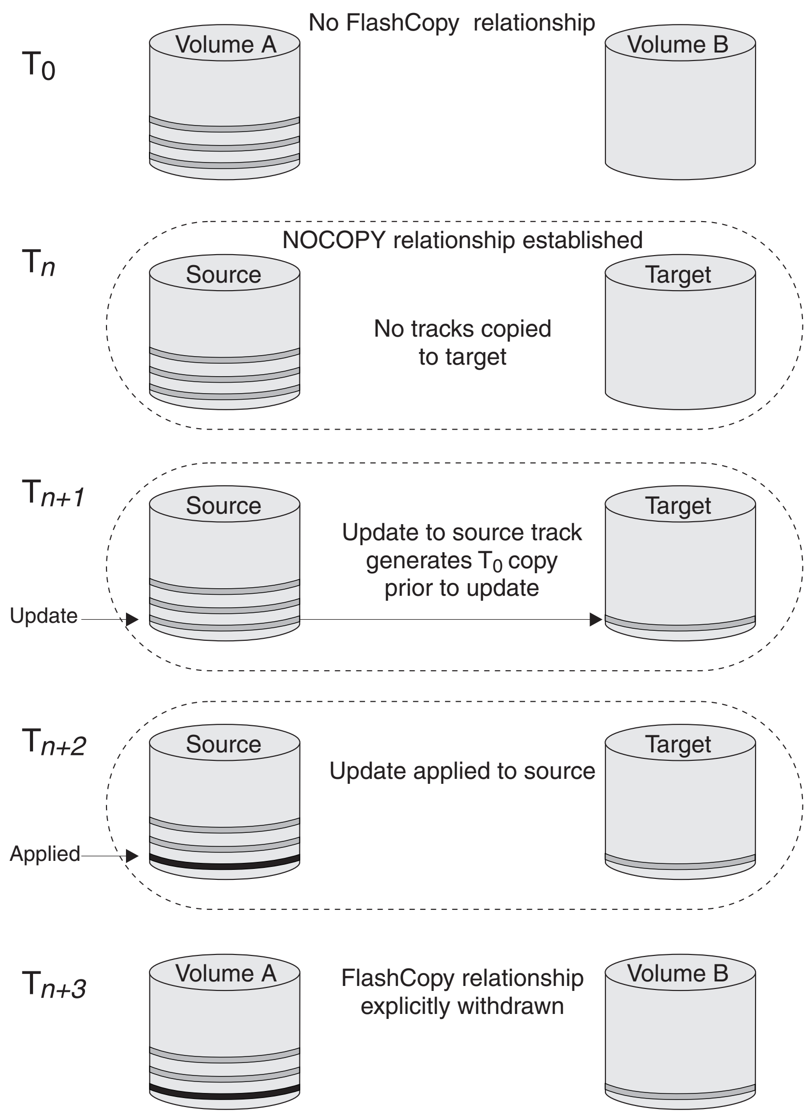


Figure 49. How the NOCOPY relationship works

When a FlashCopy NOCOPY relationship is ended, the track data on the target device is unpredictable and should not be used. If updates occur to source device tracks in the FlashCopy NOCOPY relationship, a copy of the source tracks from the point-in-time of the FlashCopy establish might not have been physically written to the target device at the time the withdraw occurs.

Note: When the target of a relationship is space efficient, the FlashCopy mode must be NOCOPY.

- **NOCOPY2COPY** changes the existing FlashCopy relationship from NOCOPY to COPY.

  When one or more NOCOPY relationships exists for a source volume, NOCOPY2COPY initiates a background copy for all target relationships with intersecting source extents from the point in time the NOCOPY was issued. Upon completion of the background copy, the converted relationship(s) are terminated.

  There must be an existing FlashCopy relationship between source and target. If one does not exist, none is created.

Note: When the target of a relationship is space efficient, do not use NOCOPY2COPY.

- **ASYNC** is only used for establishing FlashCopy relationship between PPRC secondaries and their target volumes in a Global Mirror environment.

Determining fast replication eligibility

Fast replication is a point-in-time function that allows data to be copied from a source location to a target location. FlashCopy on ESS devices and SnapShot on RAMAC Virtual Array (RVA) storage subsystems are examples of fast replication methods.

You can invoke the QFRVOLS query through the ANTRQST API to determine whether the source and target devices are capable and eligible for fast replication operations: FlashCopy V1 or FlashCopy V2 on the ESS or SnapShot on the RVA. If the QFRVOLS query indicates that you can use a volume as a fast replication target volume, then the volume can also be used as a fast replication source volume. Any condition that prevents a volume from being a fast replication target volume also prevents the volume from being a fast replication source volume.

The QFRVOLS query returns information about the type of fast replication capability for each volume or extent in a user-supplied volume list. If a fast replication operation cannot be performed, the QFRVOLS query information indicates why the volumes or extents are not eligible for fast replication operations.

The ANTRQST QFRVOLS query requires the following input to determine whether source and target volumes or extents are capable of fast replication operations, such as FlashCopy or SnapShot.

- A control volume.
- A list of target volumes (with the VOLLIST parameter) that identifies which volumes to compare to the control volume.
- A request that the volumes listed in the VOLLIST be checked for source or target volume capability for FlashCopy operations.
- An indication as to whether PPRC primaries are included as eligible target volumes.
- An indication as to whether space efficient volumes are included as eligible target volumes.

The output returns information about each volume or extent in the user-supplied volume list, which indicates whether a volume or extent can be used for fast replication.

For additional information about the parameters required for the QFRVOLS request, refer to "Subparameters for REQUEST=QFRVOLS" on page 635.

Space efficient FlashCopy

Space efficient FlashCopy refers to a FlashCopy relationship for which the target volume is a space efficient volume. With space efficient FlashCopy, the amount of physical space consumed by the target volume is limited to the minimum amount of space required to maintain the copy.

Volumes can be categorized into different types:

- Fully provisioned volumes

Fully provisioned volumes are volumes for which all of the physical space is allocated when the volume is created.

- Space efficient volumes

  Initially, repository volumes are fully provisioned volumes. For space efficient volumes, physical space is allocated on a track basis. When data is written to a space efficient volume, a track of physical space is taken from the segments assigned to a repository volume. This track of physical space is used to hold the data for the space efficient volume. A repository volume provides the physical space for multiple space efficient volumes.

When a track on the source volume of any FlashCopy relationship is updated, the current version of the track must be copied to the target device before the update can be destaged on the source device. For a space efficient FlashCopy, the current version of the track is written to the space taken from the repository volume. This track is assigned to the space efficient volume. Thus, the amount of physical space consumed by the target volume of a space efficient FlashCopy relationship is limited to the minimum amount of space required to maintain the copy.

Space Efficient FlashCopy refers to a FlashCopy relationship for which the target volume is a space efficient volume. A repository volume provides the physical space for multiple space efficient volumes.

As tracks are destaged for space efficient volumes, storage for the tracks is obtained from the segments assigned to the repository volumes. The data for a space efficient volume is stored on the repository volume, but the data is only accessible from the space efficient volume. The host does not have access to the repository volume.

For information on limitations that apply to space efficient FlashCopy, see "Limitations to space efficient FlashCopy" on page 480.

FlashCopy requirements

There are both software and hardware prerequisites for using FlashCopy.

Determining FlashCopy software requirements

All supported releases of z/OS provide FlashCopy support. FlashCopy V1 is enabled on the ESS. FlashCopy V2 is an advanced feature and requires an additional license before it can be enabled on the ESS.

Determining FlashCopy hardware requirements

FlashCopy is a feature on IBM TotalStorage ESS storage subsystems. FlashCopy operates with 3390 devices, and includes 3390 devices that are in 3380 track-compatible mode. The source and target volumes must have the same track format.

Space efficient volume capability is a licensed feature on IBM TotalStorage DS8000 storage subsystems.

FlashCopy support for open system volumes

Open System (Fixed block) devices can be specified in a FlashCopy relationship as long as a CKD volume is available (online with a UCB) and is located in the same cluster of the subsystem as the Open System device specified as the source volume. When open system volumes are managed by z/OS using FlashCopy commands, avoid using other device management methods, such as the TotalStorage ESS Copy Services Web interface, that might result in a conflict.

Making cache available

To use FlashCopy with an ESS storage subsystem, both the subsystem cache and NVS with battery backup must be active. The statuses of cache and DASD fast write have no effect on whether you can use FlashCopy. Their statuses can; however, affect the overall performance of volumes with active FlashCopy sessions.

Installing FlashCopy

Authorize FlashCopy TSO commands by adding the command names to the AUTHCMD PARM parameter of the IKJTSO*xx* member of SYS1.PARMLIB. After you have added the FlashCopy command names to the IKJTSOxx member, issue the TSO command PARMLIB UPDATE(xx) to activate the new IKJTSOxx member.

For information on FlashCopy TSO commands, including controlling access to them, refer to “Controlling access to FlashCopy commands” on page 498.

Using FlashCopy

This section describes how to use FlashCopy for various operations and how hardware and software components work together in FlashCopy operations.

Copying your data with FlashCopy

You can use the TSO commands, the ANTRQST API or the ANTTREXX program to invoke FlashCopy to perform a copy of your data.

Performing a FlashCopy V1 operation

The FlashCopy V1 release of FlashCopy in the ESS has several requirements for the devices that are involved in a FlashCopy relationship:

- The source and target devices must have the same track format and geometry.
- There can be only one FlashCopy relationship between any two devices.
- The target device must have an equal or greater number of tracks as the source device.

Due to TSO command length restrictions, you can specify one to five extent sets using the FlashCopy V1 EXTENTS keyword with the FlashCopy FCESTABL command. The number of extent sets for each relationship can be from 1 to 110, which is the maximum number of extent sets that can be used through the ANTRQST API services for FlashCopy.

Performing a FlashCopy V2 operation

If you want to create multiple copies of the same source volume or take advantage of other advanced FlashCopy enhancements, consider obtaining a license for the ESS FlashCopy V2 feature. The following list describes some of the requirements for using FlashCopy V2:

- The source and target FlashCopy volumes must reside in the same ESS, but the target volume can reside in a different LSS from the source volumes.
- A target track can reside on any eligible device in the same LSS as its associated source, including the same volume. When both source and target tracks are on the same volume, the single FlashCopy relationship restriction for a target track prevents this type of source and track relationship from using the same tracks.

- The target device must have the number of source tracks specified in the FlashCopy request, but not necessarily as many tracks as the source device.
- FlashCopy operations accommodate the relocation of tracks as long as the following specifications are met:
  - The number of specified target tracks must equal the number of specified source tracks on a given FlashCopy establish request
  - The track format and geometry of the source and target volumes must be the same. (Track format and geometry refer to 3380TCM versus 3390 format and number of tracks per cylinder.)
- Each device is allowed up to a maximum number of FlashCopy relationships. Use the FCQUERY request to determine this maximum number.
- A source device track can be in a FlashCopy relationship with up to 12 different target tracks.
- A specified target track cannot also be a source track, and a target track cannot be in more than one FlashCopy relationship.

Due to TSO command length restrictions, you can specify 1–32 extent sets using the FlashCopy V2 XTNTLST keyword with the FlashCopy FCESTABL command. When using the ANTRQST API services for FlashCopy, including invoking them through ANTTREXX, you can specify 1 to 110 extent sets.

Limitations to all types of FlashCopy

The following limitations apply to FlashCopy V1 and FlashCopy V2:

- A volume participating in a FlashCopy V1 relationship cannot accept a request to establish a FlashCopy V2 relationship for any track on that volume.
- A volume containing tracks which are participating in a FlashCopy V2 relationship cannot accept a request to establish a FlashCopy V1 relationship on that volume.
- A FlashCopy establish request fails if the specified target device or the specified target tracks are currently on a PPRC secondary.
- A FlashCopy establish request fails if the specified target device or the specified target tracks are currently on a PPRC primary, unless YES is specified in the TGTPPRIM keyword.
- A FlashCopy source device cannot be a secondary (target) device in any PPRC operation for FlashCopy V1 enabled devices.
- FlashCopy target tracks must not be established on a device that is currently acting as a target device in any of the following copy services functions: XRC, Concurrent Copy.
- A FlashCopy COPY operation might complete initialization successfully, but the background copy might not start immediately due to workload processing in the ESS.

Limitations to space efficient FlashCopy: The following limitations apply to space efficient FlashCopy:

- The source and target FlashCopy volumes must be in the same control unit. However, you can allocate the target volume in a different LSS from the source volume.
- The relationship must be full volume.
- The target device must have an equal or greater number of tracks as the source device.
- A space efficient FlashCopy relationship must be a NOCOPY relationship for the duration of the relationship.

- A source device can be the source of a FlashCopy V2 (non-space-efficient target) relationship. If both the space efficient and the V2 licenses are installed, the source device can also be the source of a space efficient relationship at the same time.
- A specified target track cannot also be a source track. A target track cannot be in more than one FlashCopy relationship.

Note: If you want to configure space efficient volumes and perform FlashCopy operations to space efficient target volumes, enable the DS8000 space efficient feature.

Creating a backup copy of your data using FlashCopy

To back up tape, use NOCOPY mode, which you request by specifying NOCOPY for the MODE option with the FlashCopy establish request. NOCOPY mode generates a logical copy of source data that can be read from another device (the target). This allows read activity (from the target device) to be separate from application read/write activity on the source device. A backup program like DFSMSdss can use the target device as input for backing up data to tape. If you need a physical copy of the source data, you can use the FlashCopy COPY mode. While the physical copy is taking place, the backup program can copy target data to tape.

Displaying information about FlashCopy volumes

The FCQUERY command displays the Copy Services status of a device. The possible Copy Services functions are XRC, PPRC, concurrent copy, and FlashCopy.

For devices which are licensed for FlashCopy V1 only, the FCQUERY command displays information only about FlashCopy. If a device is in a FlashCopy relationship, information about the other device in the FlashCopy relationship is also displayed. If information about both devices in a FlashCopy relationship is available, information about the source device is reported first, followed by information about the target device. Otherwise, only information about the requested device is reported. If a background copy is in progress on the device, the percentage of the completed background copy process is also displayed. If the device is not in any source or target FlashCopy relationship, the FCQUERY report shows the status as SIMPLEX.

The following examples show the results of using the FCQUERY command on a device that is enabled with FlashCopy V1.

```
    FCQUERY DEVN(0F4C)
```

- Issuing the FCQUERY command for device 0F4C in a V1 FlashCopy Mode COPY displays the following report:

```
ANTF0090I FCQUERY Formatted
DEVN SSID LSS CCA  CU     SERIAL    STATUS
0F4C 1010  00  0C 2105 0000000FC101 FC......4%
---- 1010  00  0D 2105 0000000FC101 FC........
ANTF0001I FCQUERY COMMAND COMPLETED FOR DEVICE 0F4C. COMPLETION CODE: 00
```

- Issuing the FCQUERY command for device 0F4C in a V1 FlashCopy Mode NOCOPY displays the following report:

```
ANTF0090I FCQUERY Formatted
DEVN SSID LSS CCA  CU     SERIAL    STATUS
0F4C 1010  00  0C 2105 0000000FC101 FC........
---- 1010  00  0D 2105 0000000FC101 FC........
ANTF0001I FCQUERY COMMAND COMPLETED FOR DEVICE 0F4C. COMPLETION CODE: 00
```

- Issuing the FCQUERY command for device 0F4C in a V1 FlashCopy in simplex (before establish or after withdraw of a single relationship:

```
ANTF0090I FCQUERY Formatted
DEVN SSID LSS CCA  CU     SERIAL    STATUS
0F4C 1010  00  0C 2105 0000000FC101 SIMPLEX...
ANTF0001I FCQUERY COMMAND COMPLETED FOR DEVICE 0F4C. COMPLETION CODE: 00
```

- Issuing the FCQUERY command for device 0F5C in V1 FlashCopy (target device query) displays the following report:

```
ANTF0090I FCQUERY Formatted
DEVN SSID LSS CCA  CU     SERIAL    STATUS
---- 1010  00  0C 2105 0000000FC101 FC........
0F5C 1010  00  0D 2105 0000000FC101 FC........
ANTF0001I FCQUERY COMMAND COMPLETED FOR DEVICE 0F5C. COMPLETION CODE: 00
```

For devices that are licensed for FlashCopy V2 or space efficient FlashCopy, the FCQUERY report shows available information about FlashCopy and other copy services relationships active on the device. If the device is not in any source or target FlashCopy relationship, the FCQUERY report shows the number of active FlashCopy relationships as 0.

The following examples show the results of using the FQUERY command on a device that is enabled with FlashCopy V2.

```
    FCQUERY DEVN(0F48)
```

- Issuing the FCQUERY command for device 0F48 with active FlashCopy relationships displays the following report:

```
ANTF0420I FCQUERY Formatted -2
DEVN SSID LSS CCA  CU     SERIAL       ACT    MAX XC PC CC RV  SEQNUM
0F48 A760  00  28 2105 0000000FCA76    263    951  N  N  N  N 00000000
ANTF0001I FCQUERY COMMAND COMPLETED FOR DEVICE 0F48. COMPLETION CODE: 00
```

- Issuing the FCQUERY command for device 0F48 in a subsystem that has FlashCopy V2 enabled, before establish or withdraw of any relationships:

```
ANTF0420I FCQUERY Formatted -2
DEVN SSID LSS CCA  CU     SERIAL       ACT    MAX XC PC CC RV  SEQNUM
0F48 A760  00  28 2105 0000000FCA76      0    951  N  N  N  N 00000000
ANTF0001I FCQUERY COMMAND COMPLETED FOR DEVICE 0F48. COMPLETION CODE: 00
```

- Issuing the FCQUERY command for device 0F48 with a PPRC Secondary relationship results in the following report:

```
ANTF0420I FCQUERY Formatted -2
DEVN SSID LSS CCA  CU     SERIAL       ACT    MAX XC PC CC RV  SEQNUM
0F48 A760  00  28 2105 0000000FCA76      0    951  N  S  N  N 00000000
ANTF0001I FCQUERY COMMAND COMPLETED FOR DEVICE 0F48. COMPLETION CODE: 00
```

  Because device 0F48 is an active PPRC Secondary, a FlashCopy establish using this as a target device is not permitted
- Issuing the FCQUERY command for a fixed block device results in the following report:

```
ANTF0420I FCQUERY Formatted -3
DEVN SSID LSS CCA  CU     SERIAL       ACT    MAX XC PC CC RV  SEQNUM
OPEN 7800  00  12 2105 000000027842      1    951  -  -  -  N 00000000
ANTF0001I FCQUERY COMMAND COMPLETED FOR DEVICE 0F40. COMPLETION CODE: 00
```

- Issuing the FCQUERY command to query a FlashCopy target volume remotely through an existing PPRC link results in a report like this:

```
ANTF0420I FCQUERY Formatted -3
DEVN SSID LSS CCA  CU     SERIAL       ACT    MAX XC PC CC RV  SEQNUM
REMT 8244  00  15 2105 000000066241      1    951  -  S  -  N 00000000
ANTF0001I FCQUERY COMMAND COMPLETED FOR DEVICE 0F40. COMPLETION CODE: 00
```

- Issuing the FCQUERY command with the SHOWRELS keyword, to show the device's relationship table information, results in a report like this:

```
FCQUERY Relationship 1
DEVN SSID LSS CCA  CU     SERIAL       ACT    MAX XC PC CC RV SE  SEQNUM
0F41 C800  00  3C 2107 000000001691      2  16699  N  P  N  N NN 00000000
RELATIONSHIP DETAIL STARTING TRACK: 00000000
DEVICE LONG BUSY FOR CG: NO  WRITE INHIBITED: NO
---------------------------------------------------
  PARTNER    SOURCE   TARGET   S F C C P C T S F P
LSS CCA SSID START    START    O V O A R R W E S M
--- --- ---- -------- -------- - - - - - - - - - -
 00  3E C800 00010001 00010001 Y N N N N N Y N Y R
     NO. OF TRACKS: 00000002  TRACKS TO COPY: 00000002
     ESTABL: 2008/12/16 20:50:36  LAST INCR:  2008/12/16 20:50:36
 00  3E C800 00010008 00010008 Y N N N N Y N Y R
     NO. OF TRACKS: 00000002  TRACKS TO COPY: 00000002
     ESTABL: 2008/12/16 20:50:36  LAST INCR:  2008/12/16 20:50:36
FCQUERY COMMAND COMPLETED FOR DEVICE 0F41. COMPLETION CODE: 00
***
```

For additional information about the FCQUERY command, refer to “FlashCopy query (FCQUERY) command” on page 510.

Inband FlashCopy

The Inband option (specified with a value of YES for the REMOTE parameter on the FlashCopy Establish request) allows FlashCopy requests to be issued remotely through an existing PPRC link. Once a FlashCopy establish is issued, the direct host connection from local to remote ESS is not required for a background copy to complete. The host connection would be needed, however, before any new FlashCopy tasks could be initiated. Inband FlashCopy can be useful if the host at the recovery site is not online.

The Inband option eliminates the need for a host connection from local to remote exclusively for FlashCopy backup. The FlashCopy request must be issued at a host processor connected to the PPRC primary volume, with the PPRC secondary volume specified as the FlashCopy source.

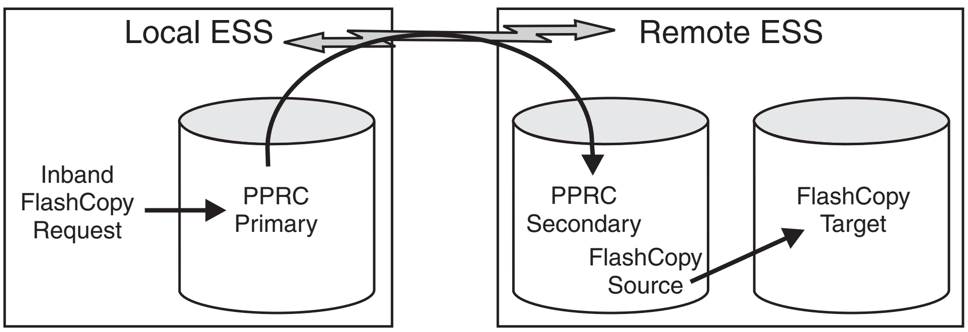


Figure 50. Inband FlashCopy Requests Are Transmitted from PPRC Primary to PPRC Secondary

All supported full-volume FlashCopy commands can be issued with the inband option, however, the THAW portion of consistency group processing is not supported with the inband option. If it is acceptable to have long busy reported on a PPRC secondary device or an XRC secondary, the FREEZE portion of consistency group processing can be issued inband and it will automatically thaw after 120 seconds. See “FlashCopy consistency groups” on page 495 for more information.

Incremental FlaschCopy

Incremental FlashCopy provides the capability to refresh a volume in a FlashCopy relationship and reduce background copy time when only a subset of data has changed. When INCREMENTAL(YES) or INCREMENTAL(YTW) is specified in a FlashCopy Establish request, the relationship between the source and target volumes is maintained (persists) after the background copy completes, and read/write operations are allowed on the source volume. The difference between specifying INCREMENTAL(YES) or INCREMENTAL(YTW) is whether the target volume is write inhibited. If you specify INCREMENTAL(YES), the FlashCopy target is write-inhibited while the incremental relationship is active. Any attempt to write to the device during this period will be failed by the hardware and surfaced as an I/O error, an error from the access method, or both. Change recording begins, in which a control bitmap is created for each volume to track changes since the previous FlashCopy was taken. When a subsequent FlashCopy establish specifying the incremental option is issued, only the data that has changed since the last incremental request is copied to the target.

As an example, let us say we have established a FlashCopy relationship with volume A as the source and volume B as the target, specifying INCREMENTAL(YES). A background copy of volume A is made on volume and, the ESS creates a control bitmap to record subsequent changes. Figure 51 on page 485

485 shows how change recording would have volumes A and B in agreement at completion of the FlashCopy operation.

Volume A	A1	A2	A3	A4	A5	A6	A7	A8	A9
Volume B	B1	B2	B3	B4	B5	B6	B7	B8	B9

Figure 51. Example of Initial FlashCopy Results

Sometime after the initial FlashCopy operation completes, tracks A1, A3, A4 and A8 are rewritten. The change recording in Figure 52 is updated to reflect this.

Volume A	A1c	A2	A3c	A4c	A5	A6	A7	A8c	A9
Volume B	B1	B2	B3	B4	B5	B6	B7	B8	B9

Figure 52. Example of Volume Disagreement

A second FlashCopy establish request is issued with the COPY and INCREMENTAL options, resulting in an update only to tracks B1, B3, B4, and B8, restoring agreement between volumes A and B.

Volume A	A1c	A2	A3c	A4c	A5	A6	A7	A8c	A9
Volume B	A1c	B2	A3c	A4c	B5	B6	B7	A8c	B9

Figure 53. Example of Incremental Update

- When an Incremental FlashCopy request includes the COPY option, and a subsequent Incremental FlashCopy is issued before the first copy is complete, then the incremental updates include those tracks that have been updated since the first copy was made, in addition to those tracks that had not yet been copied when the subsequent request was issued.
- An Incremental FlashCopy relationship cannot be converted to a non-incremental relationship. It must be withdrawn before a new non-incremental relationship can be established.
- When an incremental relationship is withdrawn, any source updates that are pending (since the last increment was applied) are not copied. The withdraw occurs immediately, even if the initial background copy has not completed, or the copy of tracks from the last increment has not completed. FCQUERY cannot show the status of either the initial background copy or an increment copy.

Incremental FlashCopy is supported only for full volumes. The normal FlashCopy restrictions apply to the target tracks of an incremental relationship.

Incremental FlashCopy Versions

When the software and microcode requirements are met, Incremental FlashCopy requests result in Incremental FlashCopy Version 2 (V2), which allows a volume to have more than one incremental relationship. With Incremental FlashCopy Version 1 (V1), a FlashCopy source can have only one incremental target.

FCQUERY requests display the version of Incremental FlashCopy with the value for CR (change recording):

N The FlashCopy establish was done without change recording.

Y The FlashCopy establish was done with Version 1 change recording.

2 The FlashCopy establish was done with Version 2 change recording. The source is able to have multiple incremental FlashCopy relationships.

In the following example, the value for CR is 2, indicating Version 2.

```
FCQUERY Relationship 1
DEVN SSID LSS CCA  CU     SERIAL       ACT    MAX XC PC CC RV SE  SEQNUM
0F40 6809  09  00 2107 0000000BDL21      1  10353  N  N  N  N NN 00000000
RELATIONSHIP DETAIL STARTING TRACK: 00000000
DEVICE LONG BUSY FOR CG: NO  WRITE INHIBITED: NO
----------------------------------------------------
  PARTNER    SOURCE   TARGET   S F C C P  C  T S F P
LSS CCA SSID START    START    O V O A R  R  W E S M
--- --- ---- -------- -------- - - - - - - - - - -
 09  01 6809 00000000 00000000 Y Y Y Y Y  2  Y N N N
     NO. OF TRACKS: 00004137  TRACKS TO COPY: 000038AF
     ESTABL: 2013/03/13 18:20:40  LAST INCR:  2013/03/13 18:20:40
FCQUERY COMMAND COMPLETED FOR DEVICE 0F40, COMPLETION CODE: 00
***
```

You can cause the system to revert to Incremental FlashCopy V1 by specifying MULTINCRFLC=NO in the DEVSUP*xx* member of parmlib.

For more information, refer to DEVSUPxx in *z/OS MVS Initialization and Tuning Reference*.

FlashCopy to PPRC primary

This FlashCopy option allows you to establish a source and target relationship in which the FlashCopy target is the primary device in a PPRC pair. In this way you can create a point-in-time copy and the make a copy of that point-in-time copy at a remote site.

A basic FlashCopy to PPRC Primary configuration, illustrated in Figure 54 on page 487, is set up as follows:

1. Establish a PPRC pair from the local (Volume B) to the remote site (Volume C) and allow the pair to reach full duplex state
2. When you want to make a point-in-time copy of volume A, establish a FlashCopy relationship with volume B (the PPRC primary) as the target, specifying MODE(COPY) and TGTPPRIM(YES). When the FlashCopy is requested, the state of the PPRC pair is changed from DUPLEX to PENDING.
3. When the background copy on volume B is complete, the FlashCopy relationship is ended, unless INCREMENTAL(YES) was specified. Resynchronization of the PPRC pair begins.
4. When the PPRC pair is resynchronized, the state returns to DUPLEX and the point-in-time copy is available at volume C.

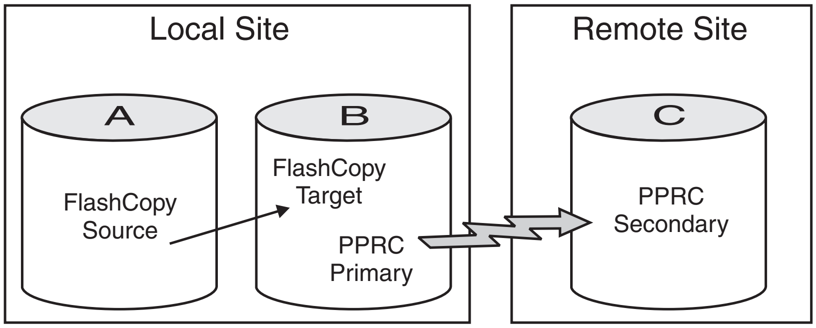


Figure 54. FlashCopy Target as PPRC Primary

The PPRC pair may be synchronous or extended distance. When PPRC-XD is used, you must ensure that the pair is synchronized before using the point-in-time copy at the remote site. If the PPRC primary is being updated, then the pair may be continually out of sync and a quiesce of write operations may be required to ensure that the remote copy is reliable.

When the FlashCopy is initiated, the PPRC pair may be in full duplex state, pending state, or suspended state. If the PPRC pair is in suspended state, then it must be resynchronized in order to obtain the remote point-in-time copy.

The FlashCopy relationship can specify a full volume or extents. When extents are specified, multiple source volumes may be copied to the PPRC primary as long as their extents do not overlap on the FlashCopy target.

The FlashCopy to PPRC Primary option can be used with the INCREMENTAL option. This results in the PPRC primary using incremental updates to resynchronize with the PPRC secondary.

Preserving mirroring

In the basic configuration illustrated by Figure 54, the PPRC (Metro Mirror) pair go into a duplex pending state while the tracks associated with the FlashCopy relationship are copied to the PPRC secondary device. With a different configuration, you can preserve mirroring, that is, keep the PPRC relationship in a full duplex state, during the FlashCopy. This is illustrated in Figure 55 on page 488.

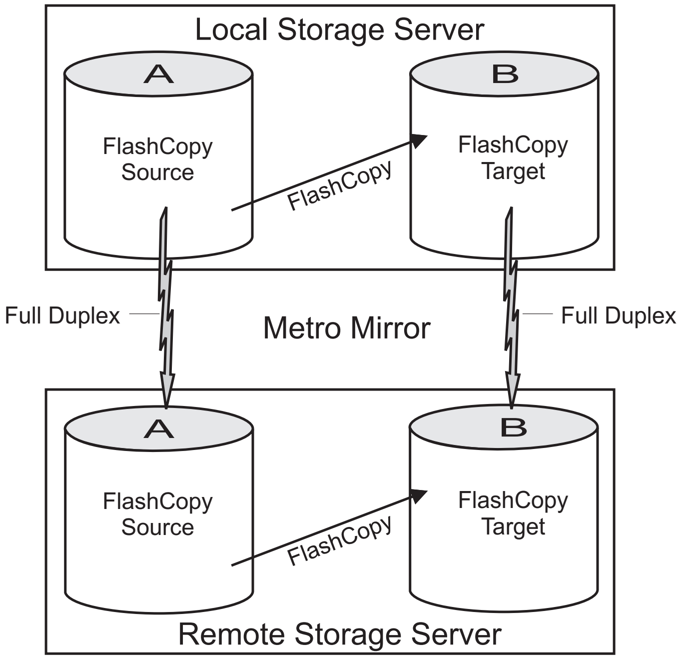


Figure 55. FlashCopy Target as PPRC Primary with Preserve Mirroring

In this illustration, there are two PPRC pairs: Local A is a PPRC primary device associated with Remote A and Local B is a PPRC primary device associated with Remote B.

The requirements for this configuration are:
- Both Local A and Local B must be Metro Mirror primary devices in full duplex mode with links established and available
- Remote A and Remote B must reside in the same Storage Facility Image (SFI)
- Local A and Local B can be the same device for an operation at the data set level
- The required microcode level must be installed in both the local and the remote storage control units

Preserve Mirror operations can be at the full volume or data set (track) level. Preserve Mirror can be used in combination with:
- Incremental FlashCopy (unless the request is attempting to take an increment using Preserve Mirror on a relationship that was established without Preserve Mirror)
- COPY, NOCOPY, or NOCOPY to COPY
- FlashCopy consistency groups
- Fixed block devices

Preserve Mirror cannot be used in combination with:
- Commit
- Revert
- Fast Reverse Restore
- Global Mirror
- Global Copy
- Cascading PPRC

Existing copy services conflicts apply, for example, the remote target cannot be a source, and Local B cannot be an XRC primary.

To establish a FlashCopy Preserve Mirror relationship, you issue a FlashCopy FCESTABL command from the Metro Mirror primary device (Local A) to another Metro Mirror primary device (Local B) with the PRESERVEMIRROR(REQUIRED) or PRESERVEMIRROR(PREFERRED) options. See “FlashCopy establish (FCESTABL) command” on page 499 for more information on the command.

Withdrawing FlashCopy relationships

Withdrawing a FlashCopy relationship differs widely between FlashCopy V1 and FlashCopy V2 with FlashCopy V2 providing more options. The FlashCopy relationship being withdrawn can be one in FlashCopy COPY mode that has not completed the background copy, or one in FlashCopy NOCOPY mode that requires that you to issue a FCWITHDR command to end the relationship.

For any other situation other than a request syntax problem (that is, parameters in error), the FCWITHDR request will be reported as successful regardless of the status of the specified tracks.

The FlashCopy withdraw process removes any existing FlashCopy relationship. The FlashCopy withdraw process locates source tracks on the source device and target tracks on the target device, and ends the FlashCopy relationship between them.

You cannot specify extent sets on the FlashCopy V1 withdraw request. Thus, a FlashCopy withdraw process defaults to a full volume withdraw operation of the specified source and target devices.

In order to withdraw a FlashCopy relationship between open (FB) devices, the CKD access device must be specified and only full volume relationships can be withdrawn.

When withdrawing a FlashCopy inband relationship, the subsystem that was specified for SSID on the PPRC establish pair command must be identified. Only full volume relationships can be withdrawn.

Note: When a FlashCopy NOCOPY relationship is ended, the track data on the target device is unpredictable and should not be used. If updates occur to source device tracks in the FlashCopy NOCOPY relationship, a copy of the source tracks from the point-in-time of the FlashCopy establish may or may not be written to the target device.

Tracks may be copied from the source to the target volume even if the source track is not changed. This includes the track that contains the volume label. Therefore, to avoid duplicate volume serial problems when the target device is later varied online, IBM recommends that you relabel the target volume after withdrawing a volume-level FlashCopy NOCOPY relationship.

Withdrawing a full-volume FlashCopy relationship

For volume FlashCopy, all FlashCopy withdraws are considered full-volume withdraws. All source tracks on the source device are removed from their FlashCopy relationship to all target tracks on the target device. Host software ensures that the volumes that are specified as source and target are the source and target devices, respectively, in an active FlashCopy relationship.

Anytime that a FlashCopy relationship is withdrawn, the data on the target volume might be left in an unpredictable state. Figure 56 shows how data on the target volume might be inconsistent with the source volume after a FCWITHDR command has been processed. It shows a typical backup cycle that starts with a complete backup on a Saturday night, updates to the source volume during the first three days of the week, and a point-in-time backup on a Wednesday night.

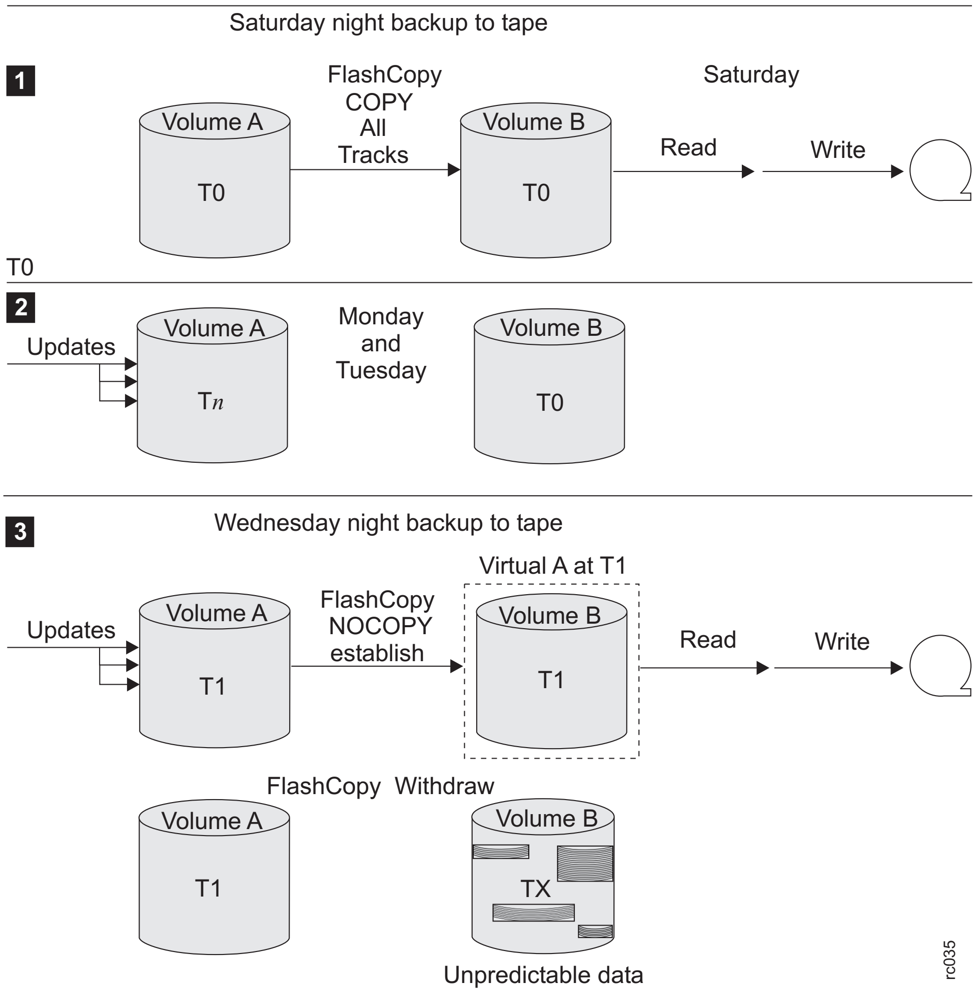


Figure 56. The state of a target volume after a FCWITHDR command processes

The example presumes a typical customer backup cycle with the following assumptions:

1 On Saturday, at the end of a business week, the customer wants to do a full backup. A FlashCopy MODE(COPY) relationship is established between volume A and volume B. A background copy process starts and all tracks are copied from volume A to volume B (a T0 copy). After the FlashCopy relationship is established, read access from the target copy is immediately possible. As part of the backup cycle the customer uses DFSMSdss to create a backup copy from volume B to tape. By creating the tape backup copy from volume B, the customer gets the volume A data from the point-in-time of the establish and eliminates I/O contention with application programs continuing to run on volume A. When the background copy finishes the FlashCopy relationship is removed.

NOTE: The data on volume B is unpredictable in the case where a FlashCopy MODE(COPY) relationship is withdrawn before the background copy to volume B completes.

2 On Monday and Tuesday, application programs continue to make updates to volume A. The backup copy that was created on tape on Saturday from volume B is a virtual copy of volume A at the FlashCopy establish time on Saturday. Updates that continue on volume A after the FlashCopy relationship was established are not included on volume B. Consequently, the data on volume B is no longer consistent with the data on volume A.

3 Application updates continue on volume A. A new FlashCopy establish with MODE(NOCOPY) is done Wednesday night in preparation for another tape backup cycle. The new point-in-time data is now accessible from B, and the tape backup copy can be made from volume B. Since this establish was done using MODE(NOCOPY), the state of the physical data on volume B at this point is unpredictable. Reading from volume B while the relationship is established will return the point-in-time data from volume A, while some tracks may also have been physically copied from volume A to volume B during this time. After the tape backup copy is complete, a FlashCopy withdraw command removes the FlashCopy relationship between volume A and volume B. At this time the contents of volume B are unreliable and should not be used.

Withdrawing a Preserve Mirror FlashCopy relationship

FlashCopy Withdraw processing for a Preserve Mirror relationship is designed to minimize differences between the local and remote FlashCopy targets (Local B and Remote B in Figure 55 on page 488), difference that can occur, even though the PPRC pair is full duplex at all times, because the local and remote storage servers perform the background copy for their respective relationships in a different manner or at a different pace. Note that even without PPRC, the content of target tracks is undetermined after a FlashCopy Withdraw.

For a FlashCopy Preserve Mirror relationship:

- The FCWITHDR command and the FCWITHDRAW request withdraw the mirrored relationship immediately at both the local and the remote storage servers, unless you specify the DDSW option. An indicator that the target PPRC pair was the target of a withdrawn preserve mirror relationship is set, so that this can be reflected in the results of a CQUERY command. A FlashCopy Withdraw issued without that option to the local FlashCopy source or target (Local A or Local B) starts a background copy for both the local and remote relationships.
- The DDSW option, when used with a FlashCopy Withdraw issued to the remote FlashCopy source (Remote A), starts a background copy for the remote relationship. A FlashCopy Withdraw with DDSW issued to the remote FlashCopy target (Remote B) will be rejected.
- The FCWITHDR command and the FCWITHDRAW request issued to either the remote FlashCopy source or target (Remote A or Remote B) withdraw only the remote relationship, and set an indicator that there may be tracks that are not identical only on the remote device.

FlashCopy V2 withdraw options

For FlashCopy V2, all FlashCopy withdraws are considered track extent withdraws.

Withdraw processing, within the specified tracks, removes the active FlashCopy relationship between source tracks on the source device and target tracks on the target device.

Withdraw target only: Specification of the TDEVN parameter without the SDEVN parameter allows the user to withdraw all FlashCopy relationships which have target extents on the specified TDEVN, regardless of what source volumes are involved in those FlashCopy relationships. In this case, withdraw processing (in the ESS) identifies where the source tracks are (volume and track location) and removes all information about the relationships from all devices involved.

Withdraw source and target: Specification of the both the SDEVN parameter and the TDEVN parameter allows the user to limit the scope of the withdraw to those FlashCopy relationships which have source extents on the specified SDEVN and corresponding target extents on the TDEVN, regardless of whatever source and target relationships with other devices may exist.

FlashCopy Version 1 withdraw specifying the source device and the target device is different from withdraw target only in one respect: if the source device is not the source device in a relationship with the target device, the request fails. FlashCopy Version 2 withdraw specifying the source device and target device affects existing FlashCopy relationships with the same devices in the same relationship order (source extents on the source device, target extents on the target device). If a FlashCopy relationship is within the specified extents but is not in the same relationship order as the withdraw request, the existing FlashCopy relationship is not changed. FlashCopy relationships to devices not specified in the withdraw request are unchanged.

Using DDSW(YES): The FCWITHDR command with the deleted data space withdraw (DDSW) parameter, when issued against an entire source volume, removes all target extents on the specified source device from their associated FlashCopy relationships. In addition, any FlashCopy NOCOPY source tracks on the specified source device are changed to COPY (background copy) source tracks. This process causes all source tracks to be copied to their respective target tracks and all FlashCopy relationships to be removed from the specified source device. For additional information about how to restrict DDSW processing to selected portions of a volume, see "Limiting the scope of a withdraw using the XTNTLST" on page 493.

A FlashCopy withdraw with DDSW(YES) issued to the source of a relationship will cause a background copy to be initiated. It is not allowed if there is a space efficient target volume. The background copy will be initiated for any source relationships where the target is a fully provisioned volume, and then the command will receive a failing return code or error message because of the space efficient relationship.

The main benefit of the DDSW(YES) parameter is that it can be used to easily free up tracks on both the source and corresponding target volumes that are in existing FlashCopy relationships which are no longer needed because the original source data justifying the relationship has been deleted. Another example might be that prior to starting a backup cycle using DFSMSdss, you might want to make sure that all relationships have been cleaned up on the subject source volume.

However, if there is a space efficient target volume, you can not issue a FlashCopy withdraw with DDSW(YES) to the source of a relationship because the withdraw initiates the background copy.

Figure 57 illustrates the sequence of events that occur when the ESS removes target FlashCopy relationships that result when a FCWITHDR with DDSW(YES) command processes.

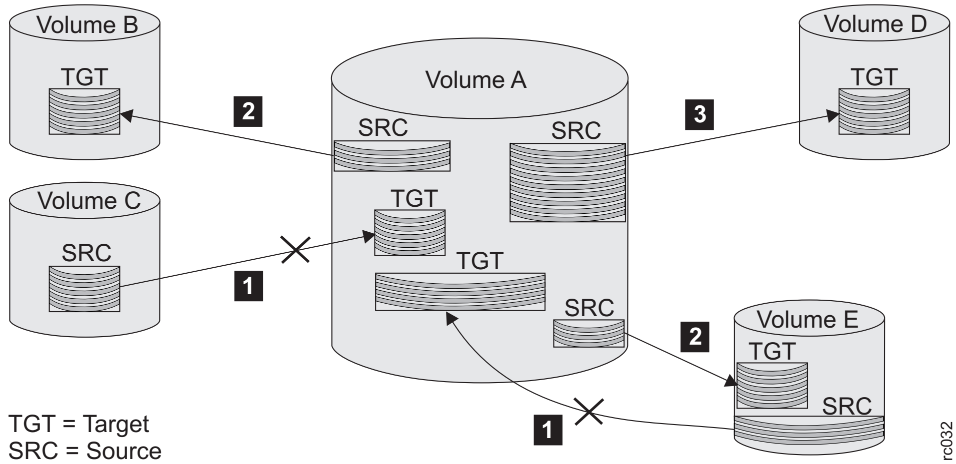


Figure 57. How the ESS processes the FCWITHDR with DDSW(YES) command

Example assumptions:

- FlashCopy NOCOPY relationships are established with source tracks on Volume A and target tracks on Volume B and Volume E.
- A FlashCopy COPY relationship is established with source tracks on Volume A and target tracks on Volume D.
- FlashCopy relationships are established with source tracks on Volume C and Volume E, and target tracks on Volume A.

When this command is processed against Volume A:

```
FCWITHDR SDEVN(ssss) DDSW(YES)
```

the following events occur:

1 All target tracks on the source volume A are removed from their associated FlashCopy relationships (volume C and volume E).

2 For FlashCopy relationships that were established between the source volume A and target volumes (volume B and volume E) using the NOCOPY mode, the following occurs:

1. All FlashCopy source tracks are changed to COPY (background copy) source tracks.
2. The tracks are physically copied to their respective target tracks.
3. When the background copies are complete, all FlashCopy relationships are removed.

3 A FlashCopy relationship that was established between the source volume A and the target volume D using the COPY mode is allowed to complete. The FlashCopy relationship is removed automatically when the copy completes.

Limiting the scope of a withdraw using the XTNTLST: XTNTLST acts in conjunction with the other withdraw techniques such as target only, source and target, and DDSW, to restrict which relationships are affected by the withdraw command processing. The XTNTLST becomes an additional filter to determine which relationships and tracks are processed.

If DDSW(YES) is not specified, this is considered a partial withdraw. If an existing relationship intersects the specified extent ranges, only those tracks within the specified ranges will be withdrawn, and an extent split will occur to create one or more new relationships for the remaining tracks not being withdrawn. The request could fail if the additional relationship(s) cause you to go beyond the maximum number of relationships allowed.

If DDSW(YES) is specified, handling of the extent specification is somewhat different. If an existing source relationship intersects a specified extent range, that entire relationship will be converted to a copy relationship and withdrawn once the background copy has completed, even if the relationship contains tracks which are not within the specified range. If an existing target relationship intersects a specified extent range, only those tracks within the specified range will be withdrawn, and an extent split will occur to create one or more new relationships for the remaining tracks not being withdrawn.

The following provides an example of how you can limit the scope of a DDSW(YES) withdraw using the XTNTLST parameter. The basis for this example is Figure 57 on page 493. The processing that occurs is based on the issuance of the following command and supporting assumptions.

Example assumptions:

- The extent range c1h1 through c2h2 includes the last 5 tracks from the source extent for the COPY relationship between Volume A and Volume B.
- The extent range c1h1 through c2h2 includes all tracks from the source extent for the COPY relationship between Volume A and Volume D.
- The extent range c1h1 through c2h2 includes all tracks from the target extent for the relationship between Volume C and Volume A.
- The extent range c1h1 through c2h2 includes the first 37 tracks from the target extent for the relationship between Volume E and Volume A.

The following events occur when the command FCWITHDR SDEVN(*ssss*) DDSW(YES) XTNTLST (c1h1 c2h2 c3h3 c4h4) is processed against Volume A:

- All target tracks on volume A in the relationship with volume C are removed from that relationship. Only target tracks within the specified c1h1 through c2h2 extent range on volume A in the relationship with volume E are removed from that relationship.
- For the FlashCopy relationship that is established using the NOCOPY mode between volume A and volume B, the following occurs:
  1. All FlashCopy source tracks in this relationship are changed to COPY (background copy) source tracks, even though some of those tracks lie outside of the specified c1h1 through c2h2 extent range.
  2. The tracks are physically copied to their respective target tracks.
  3. When the background copy is complete, all tracks are removed from that FlashCopy relationship.
- For the FlashCopy relationship that is established using the COPY mode between volume A and volume D, all tracks lie within the specified c1h1 through c2h2 extent range. The relationship does not need to be converted from

NOCOPY to COPY, and the existing background copy task is allowed to complete and tracks removed from that relationship at that time (normal behavior).

- For the FlashCopy relationship that is established using the NOCOPY mode between volume A and volume E, all tracks lie outside of the specified c1h1 through c2h2 extent range. The relationship is not affected by processing for this command, and continues to exist in NOCOPY mode.

For additional information about the XTNTLST parameter, refer to page "XTNTLST" on page 506.

FlashCopy consistency groups

FlashCopy is often used to make copies of data that frequently crosses the volume boundary. In such cases, the data that crosses the volume boundary needs to be consistent, that is, it must be copied in the proper order. FlashCopy Consistency groups, available in the System Data Mover API, provide a mechanism for achieving a consistent data copy across multiple volumes without requiring that application I/O be quiesced.

In the case of production data, application impact must be minimized. Without consistency groups, you must first quiesce the application, establish the FlashCopy relationships, and then restart the application. This process can be very disruptive, causing application outages or data unavailability for an unacceptable period of time.

For PPRC or XRC, the process of creating consistent point-in-time copies at a secondary site can be equally disruptive. And, during the process, data is not being copied to the secondary site. In general, the process consists of the following tasks:

1. FREEZE or SUSPEND all established pairs
2. FlashCopy the secondary volumes
3. Reestablish FlashCopy relationships
4. Reestablish links/paths for all pairs
5. Reestablish and RESYNC all pairs

The FlashCopy consistency group function improves this process for data-consistent copies or dependent write-consistent copies using both PPRC and XRC. Consistency group formation is initiated when you specify FREEZE in the ACTION parameter with a FlashCopy Establish request. When this request is received, the ESS gives a long busy to the specified volumes, stalling write I/O completion to the FlashCopy source. Once the Establish request has been issued for all participating FlashCopy relationships, issue a FlashCopy Withdraw request specifying THAW in the ACTION parameter to each LSS containing participating volumes so that write I/O can resume. Write I/O automatically resumes after 120 seconds even if the FlashCopy Withdraw THAW request is not issued.

Note: The two-inute timer can be adjusted using the ESS Specialist GUI. The same timer is used for both FlashCopy and PPRC, so adjusting this timer will affect both processes.

The freeze operation affects the entire volume. Even if the FlashCopy Establish request contains extent specifications, the entire volume will remain busy to host operations, but the FlashCopy relationship between the source and target will be established as specified using whatever parameters were issued with the request.

Since the request to form a consistency group is processed by the LSS for all volumes in the LSS, multiple consistency groups with volumes in the same LSS must not be formed at the same time.

If a FlashCopy Withdraw request with THAW specified is received by an LSS that does not have any volumes in a consistency group being formed, the command is accepted, but no actual processing takes place.

Since the command to end a consistency group is processed at the LSS level, the target device information specified via the ANTRQST API or via the TSO command only needs to contain valid hexadecimal values for parsing purposes.

Note: The THAW portion of consistency group processing is not supported with the inband option (FlashCopy established with YES specified for REMOTE parameter). If it is acceptable to have long busy reported on a PPRC secondary device or an XRC secondary, the FREEZE portion of consistency group processing can be issued inband and it will automatically thaw after 120.

Combining copy services functions with FlashCopy

You can combine copy services functions such as PPRC and XRC with FlashCopy as long as they are enabled on the ESS. You can make a FlashCopy of a PPRC or XRC primary or secondary volume.

For additional information about:

- Using PPRC with FlashCopy, refer to "Using PPRC with FlashCopy" on page 353.
- Using XRC with FlashCopy, refer to "Using XRC with FlashCopy and tertiary volumes" on page 182.
- Determining allowable combinations of FlashCopy operations on the same volume, refer to "Combining copy services operations" on page 7.

Using space efficient FlashCopy

Use space efficient FlashCopy for copies that are short term, such as copies that are backed up to tapes. You can also use space efficient FlashCopy for copies that are kept long term, if there are few updates to the source and target volumes. Space efficient FlashCopy relationships are limited to full volume relationships. Thus, extent specifications are not supported.

You must specify NOCOPY for space efficient relationships. You can use space efficient volumes with all other FlashCopy functions, such as change recording and consistency groups.

Copying your data with space efficient FlashCopy

You can use the SETGTOK keyword on the FCESTABL TSO command and on the FCESTABLISH request (with ANTRQST or ANTTREXX) to indicate whether the target can be a space efficient volume. The default does not allow the target to be a space efficient volume. The relationship must be full volume.

Exhausted space situations

When space on the repository volume is completely exhausted, the space efficient volume becomes unavailable. If space is exhausted while a relationship exists, one of the two following situations occur:

- The relationship is put in a failed state, which means the target copy becomes invalid. Writes continues to the source volume.
- The copy is preserved by inhibiting updates to the source volume. This behavior only occurs when the space efficient FlashCopy relationship is part of a global mirror session.

In a Global Mirror environment, when physical space is exhausted, writes to the FlashCopy source volume (PPRC secondary volume) are inhibited. The space exhaustion causes the PPRC relationship to suspend, and the Global Mirror consistency group processing to fail. The suspension preserves the consistency prior to the space efficient space failure. Thus, the PPRC secondary volumes can resume back to the last consistent state in the case of an outage. While you can use space efficient volumes for XRC or PPRC secondaries, it could deplete the available storage in the repository, resulting in the suspension of their mirror environment.

Releasing space for space efficient volumes

Physical space for a space efficient volume can be released in the following ways:
- Initialize the volume using ICKDSF INIT.
- Release space during FlashCopy establish processing if the target volume is a space efficient volume and use the establish operation to create a new relationship.
- Specify the space release option on an FCWITHDRAW request (with ANTRQST or ANTTREXX), or with an ICKDSF FLASHCPY WITHDRAW request.

Note:
1. The volume space is not released if the following situations occur:
   - You choose the the ICKDSF INIT option.
   - The initialized volume is a space efficient volume.
   - The volume is also currently the source of a FlashCopy relationship.
2. If you use the space release option, all space associated with the space efficient volume is returned to the repository, and the volume is not initialized. Do not use the space release option until ICKDSF INIT is run against the device. Use the ICKDSF INIT to release space associated with a volume. The allocated space is released asynchronously so may not be reported immediately.

Using FlashCopy TSO commands

IBM TotalStorage ESS FlashCopy functions can be invoked by the commands in Table 59:

Table 59. FlashCopy commands

Command . . .	Description . . .	See . . .
FCESTABL	Establishes a FlashCopy relationship.	"FlashCopy establish (FCESTABL) command" on page 499
FCQUERY	Queries an ESS device.	"FlashCopy query (FCQUERY) command" on page 510
FCWITHDR	Withdraws a FlashCopy relationship.	"FlashCopy withdraw (FCWITHDR) command" on page 513

The commands listed in the table are authorized commands. They should be protected by RACF facility class protection or other authorization methods. These commands should be protected in a way similar to the PPRC commands.

Controlling access to FlashCopy commands

The following are two ways to protect access to FlashCopy commands:

- Put the commands in a library protected by RACF.
- Define resource profiles in the RACF Facility class and restrict access to those profiles.

Steps for placing TSO commands in a library protected by RACF

To restrict FlashCopy TSO commands to authorized storage administrators only, place the commands in a library protected by RACF.

Perform the following steps to place FlashCopy commands in a library that RACF protects:

1. Issue the following RDEFINE command for each FlashCopy command that you want defined to RACF:

   ```
   RDEFINE PROGRAM cmdname ADDMEM('SYS1.CMDLIB') UACC(NONE)
   ```

   where:

   cmdname
   : Defines the FlashCopy command name. Issue a separate RDEFINE command for each of the FlashCopy commands. RACF can only perform checking on commands that are defined to it.

2. Issue the PERMIT command for all commands and authorized FlashCopy TSO command users as follows:

   ```
   PERMIT cmdname CLASS(PROGRAM) ID(name) ACCESS(READ)
   ```

   where:

   cmdname
   : Defines the FlashCopy TSO command name.

   name
   : Defines the user ID receiving RACF access authority for that command name.

3. Issue the SETROPTS command from a user ID having the appropriate authority:

   ```
   SETROPTS CLASSACT(PROGRAM) WHEN(PROGRAM) REFRESH
   ```

Defining resource profiles in the RACF Facility class

You can limit the use of FlashCopy commands by defining resource profiles in the RACF Facility class and restricting access to these profiles. To use a protected command, you need read-access authority to the applicable profile. For FlashCopy commands, there are two RACF Facility class resource profile names:

- STGADMIN.ANT.ESFC.COMMANDS

- STGADMIN.ANT.ESFC.FCQUERY

Note: Authorize FCQUERY command use with the STGADMIN.ANT.ESFC.COMMANDS profile or the STGADMIN.ANT.ESFC.FCQUERY profile. FlashCopy first checks STGADMIN.ANT.ESFC.COMMANDS for authorization. If authorization is not permitted with STGADMIN.ANT.ESFC.COMMANDS, FlashCopy checks the STGADMIN.ANT.ESFC.FCQUERY profile for authorization to issue the FCQUERY command.

Examples: The following RACF command examples activate the RACF FACILITY class, define the profile for the FlashCopy commands, and give user STGADMIN authority to use this profile:

1. Activate the RACF FACILITY class:
   - SETROPTS CLASSACT(FACILITY)
2. Define the profile for FlashCopy commands, and authorize user STGADMIN to use this profile:
   - RDEFINE FACILITY STGADMIN.ANT.ESFC.COMMANDS UACC(NONE)
   - PERMIT STGADMIN.ANT.ESFC.COMMANDS CLASS(FACILITY) ID(STGADMIN) ACCESS(READ)

For additional information about activating the RACF facility class, and how to define and authorize users to the FlashCopy command profiles, refer to *z/OS Security Server RACF Security Administrator's Guide*.

FlashCopy establish (FCESTABL) command

Use the FCESTABL command to establish a FlashCopy relationship between the user-specified source and target devices. The FlashCopy operation runs as an optional background copy process. FlashCopy relationship begins when the FlashCopy is initiated and ends when the background copy completes.

FCESTABL command syntax

The syntax of the FCESTABL command is:

```
►►──FCESTABL──────────────────────────────────────────────────────────────────────►

►──┬─SDEVN(ssss)──┬──────────────────┬──TDEVN(tttt)──┬─────────────────────┬─┤ Extents ├─┬──►
   │              └─SUBCHSET(subchset)┘               └─TSUBCHSET(tsubchset)┘             │
   └─DEVN(devno)──┬──────────────────┬──SOURCE(srcid)──TARGET(tgtid)─┤ Open and Remote ├──┘
                  └─SUBCHSET(subchset)┘

►──┬────────────────────────┬──┬──────────────────────────────┬──┬────────────────────────┬──►
   │              ┌─NO──┐   │  │          ┌─COPY────────┐     │  │              ┌─NO──┐   │
   └─ONLINTGT──(──┴─YES─┴─)─┘  └─MODE──(──┼─NOCOPY──────┼──)──┘  └─TGTPPRIM(──┴─YES─┴─)─┘
                                          ├─NOCOPY2COPY─┤
                                          └─ASYNC───────┘

►──┬─────────────────────┬──┬─────────────────────────────┬──┬───────────────────────────┬──►
   │          ┌─YES─┐    │  │                ┌─NO──┐      │  └─ACTION(──┬─FREEZE─┬──)──┘
   └─TGTUCB(──┴─NO──┴─)──┘  └─INCREMENTAL(──┼─YES─┼──)───┘             └─FRR────┘
                                            └─YTW─┘

►──┬────────────────────────┬──┬──────────────────────────────────────────────────┬──►◄
   │             ┌─NO──┐    │  │                         ┌─NO─────────────┐       │
   └─SETGTOK──(──┴─YES─┴─)──┘  └─┬─PRESERVEMIRROR─┬──(──┼─┬─REQUIRED──┬─┼──)────┘
                                 └─PRESMIR────────┘      │ └─REQ───────┘ │
                                                         └─┬─PREFERRED─┬─┘
                                                           └─PREF──────┘
```

Extents:

```
├──┬─────────────────────────────────────────────────────┬──────────────────────────┤
   │                ┌─,───────────┐                      │
   ├─EXTENTS──(────▼─c1h1──c2h2──┴──)───────────────────┤
   │                ┌─,───────────────────────────┐      │
   ├─XTNTLST──(────▼─c1h1──c2h2──c1h1──c2h2──────┴──)──┤
   └─XXTNTLST──(──c1h1──c2h2──c1h1──c2h2──)─────────────┘
```

Open and Remote:

```
├──┬─OPENDVCS(YES)──REMOTE(YES)──────────────┬──────────────────────────────────────┤
   ├─OPENDVCS(YES)──REMOTE(NO)───────────────┤
   └─OPENDVCS(NO)──REMOTE(YES)──SSID(ssid)──┘
```

Required parameters

SDEVN

Specifies the hexadecimal device number of a local CKD device that will be used as the FlashCopy source. SDEVN is not supported when OPENDVCS(YES) or REMOTE(YES) is specified.

TDEVN

Specifies the hexadecimal device number of a local CKD device that will be used as the FlashCopy target. This number can be entered without the 'X' designation. The target device can be online or offline. If the target device is online, the completion of the FCESTABL command depends on the setting of the ONLINTGT parameter.

Note:

1. *Online* means not only online to the host from which you are operating, but potentially online to any other host attached to the ESS. If the device is in a path group to another host, it is considered *online*.
2. TDEVN is not supported when OPENDVCS(YES) or REMOTE(YES) is specified.

DEVN

When OPENDVCS(YES) is specified for a local FlashCopy relationship (REMOTE(NO) is specified or allowed to default), DEVN must specify a CKD access volume located in the same subsystem cluster as the fixed block device identified by SOURCE in this command.

When OPENDVCS(YES) is specified with (REMOTE(YES), DEVN must specify a CKD access volume located in the same subsystem cluster as the PPRC primary device that is paired with the PPRC secondary specified as the FlashCopy source.

When REMOTE(YES) is specified for CKD volumes (OPENDVCS(NO) is specified or allowed to default), DEVN must specify the PPRC primary device that is paired with the PPRC secondary specified as the FlashCopy source.

DEVN must be a 4-digit hexadecimal address of a configured device with a UCB on the System z system issuing the command.

SOURCE

When OPENDVCS(YES) or REMOTE(YES) is specified, SOURCE specifies the source volume's storage control serial number (*serialno)*, logical storage subsystem (*lss*), and channel connection address (*cca*) or logical unit number (*lun*).

Where:

serialno

Specifies the storage control serial number that can include up to 10 digits, depending on the type of storage control.

lss

Specifies the two-digit hexadecimal value for the logical subsystem (LSS) for the device (ESS only).

Note: The LSS number is required if the storage control supports logical subsystems (like the ESS) and not allowed if the storage control does not support logical subsystems.

lun **or** *cca*

lun specifies the two-digit hexadecimal value for the logical unit number of the open device. *cca* specifies the two-digit hexadecimal channel connection address

TARGET

When OPENDVCS(YES) or REMOTE(YES) is specified, TARGET identifies the target volume in the FlashCopy relationship. The descriptions for these values are the same as those shown for the SOURCE parameter.

Optional parameters

OPENDVCS

Specifies whether Open System (fixed block) devices or CKD devices are addressed by this command. The values are:

NO The required parameters (SDEVN and TDEVN) identify CKD devices. The default is NO.

YES
Specifies that the DEVN, SOURCE, and TARGET parameters identify fixed block devices using a CKD access device.

SSID
When REMOTE(YES) is specified for CKD volumes (OPENDVCS(NO) is specified or allowed to default), this is the 4-digit hexadecimal address of the storage subsystem where the remote FlashCopy is located. This must be the same value as that specified for SSID of the secondary device on the PPRC CESTPAIR command.

REMOTE
Specifies whether the request is for a subsystem that is not directly attached to the issuing processor. The values are:

NO The request is directed to a device on a subsystem locally attached to the issuing processor. The default is NO.

YES
The request is directed to a remote subsystem using Inband subsystem functions. When REMOTE(YES) is specified, the request must be for a full volume.

When specifying an Inband request for a CKD secondary device, DEVN must identify a PPRC primary device in a subsystem accessible by the host processor issuing the request, and the device specified as the FlashCopy source must also be the PPRC secondary for the DEVN device. The subsystem id must be specified with the SSID parameter as well.

When specifying an Inband Flashcopy request for an open (fixed block) device, DEVN must identify an online CKD System z access device located in the same subsystem cluster as the primary of a fixed block PPRC pair, with the secondary designated as FlashCopy target. The device specified as the FlashCopy Source device must be the PPRC secondary of a FB PPRC pair.

Note: REMOTE(YES) is not valid when the EXTENTS or XTNTLST parameters are specified.

INCREMENTAL
Specifies whether the FlashCopy establish relationship remains active after initial copy is complete allowing subsequent changes to be tracked. Thus, future FlashCopy operations only require a subset of the volume to be copied. The values are:

NO Indicates that the FlashCopy relationship ends after the background copy has completed. The default is NO.

YES
Indicates that the FlashCopy relationship remains in effect after the request completes. Subsequent changes are tracked so that future FlashCopy operations are performed incrementally. This relationship continues until explicitly terminated with a FlashCopy Withdraw request.

The FlashCopy target is write-inhibited while the incremental relationship is active. Any attempt to write to the device during this period fails with the hardware and surfaced as an I/O error, an error from the access method, or both.

YTW

Indicates that the FlashCopy relationship remains in effect after the request completes. Subsequent changes are tracked so that future FlashCopy operations are performed incrementally. This relationship continues until explicitly terminated with a FlashCopy Withdraw request.

Note: The FlashCopy target is writable while the incremental relationship is active. Any writes done to the target during this period are overwritten if a subsequent increment is done, keeping the target a true copy of the source. If the relationship is reversed, the changes made to the target are reflected on the source.

TGTPPRIM

Specifies whether the target in the FlashCopy relationship can be the primary in a PPRC pair. The values are:

NO The target in this FlashCopy relationship cannot be the primary in a PPRC pair. The default is NO.

YES

The target in this FlashCopy relationship can be the primary in a PPRC pair. This FCESTABL request proceeds normally to the specified target but the hardware ignores the PPRC status of the target.

Note: If FRR is specified, set **YES** for TGTPPRIM.

TGTUCB(YES|NO)

Indicates whether an MVS device number is being used for target addressability.

YES

Indicates that the SDEVN and TDEVN keywords are being used to identify the source and target devices in the relationship using MVS device numbers. The default is YES.

NO Indicates that an MVS device number will not be used for the target device (TDEVN keyword will not be used). Instead, the SDEVN and TARGET keyword will be used.

Specifying TGTUCB(NO) will be ignored when specified with REMOTE(YES) or OPENDVCS(YES). This is because these two types of requests already use the TARGET keyword instead of the TDEVN keyword and the system does not have direct addressability to the target volume in the relationship.

MODE

Specifies the type of FlashCopy operation. For more information about these copy modes, see "Background copy of FlashCopy tracks" on page 475.

One of the following subparameters must accompany the use of the MODE parameter:

COPY

Specifies that a FlashCopy relationship be established between the source and target devices for the specified extent ranges. It also specifies that a background copy of all source tracks be made. The FlashCopy relationship ends when the background copy is complete or the FCWITHDR command is issued.

If you specify MODE(COPY), data on the target device is overlaid within the track extents specified. The default is COPY.

NOCOPY
Specifies that a FlashCopy relationship be established between the source and target devices for the specified volume or extent ranges without initiating a background copy operation. When the ESS receives an update to a source track in a FlashCopy NOCOPY relationship, a copy of the point-in-time (pre-update) data is preserved on the target volume.

A FlashCopy relationship established in NOCOPY mode remains active until one of the following occurs:

- All specified source device tracks are updated by user applications.
- The ESS copies all tracks from source to target when a threshold number of source tracks are updated.
- A FlashCopy withdraw (using the FCWITHDR command) is issued to remove the FlashCopy relationship.
- A NOCOPY relationship is converted to a COPY relationship through the NOCOPY2COPY function.

Note:

1. When a FlashCopy NOCOPY relationship is ended, the track data on the target device is unpredictable and should not be used. If updates occur to source device tracks in the FlashCopy NOCOPY relationship, a copy of the source tracks from the point-in-time of the FlashCopy establish may or may not be written to the target device.

   Tracks may be copied from the source to the target volume even if the source track is not changed. This includes the track that contains the volume label. Therefore, to avoid duplicate volume serial problems when the target device is later varied online, IBM recommends that you relabel the target volume after withdrawing a volume-level FlashCopy NOCOPY relationship.
2. There is no host system data protection for the track extent areas on the target device like the host system protection for the source device data.

NOCOPY2COPY
Starts a background copy for source NOCOPY relationships intersecting with the extents specified. The relationships end when the background copy is completed. In a Preserve Mirror relationship, FCESTABL with NOCOPY2COPY issued to the local FlashCopy source starts a background copy for both the local and the remote relationships; issued to the remote FlashCopy source, it starts a background copy for the remote relationship only. See “Preserving mirroring” on page 487 for more information.

ASYNC
Specifies that this FlashCopy relationship is being established to a volume set that is to be part of a Global Mirror session. When ASYNC is specified, INCREMENTAL=YES and TGTPPRIM=YES are ignored.

ONLINTGT
Specifies whether an online device can be used as the target of a FlashCopy relationship.

One of the following subparameters must accompany the use of the ONLINTGT parameter:

YES
Specifies that the FlashCopy relationship can be established even if the target device appears to be online. If this option is selected for a full-volume establish with background copy , after the background copy

completes, the target volume must be varied offline and then back online before the new volume serial and VTOC on the target volume can be recognized and data on the target volume accessed.

NO Specifies that the FlashCopy relationship cannot be established if the target device appears to be online. The default is NO.

Note:

1. *Online* means not only online to the host from which you are operating, but potentially online to any other host attached to the ESS. If the device is in a path group to another host, it is considered *online*.
2. ONLINTGT is not supported when OPENDVCS(YES) is specified.

EXTENTS, XTNTLST, or XXTNTLST

Specifies the location of tracks (extents) on the source and target devices to associate in a FlashCopy relationship. You can specify EXTENTS if FlashCopy V1 is enabled, or you can specify either EXTENTS or XTNTLST if FlashCopy V2 is enabled. These parameters are explained later. If you do not specify either of these parameters or a track extent, the FlashCopy FCESTABL command defaults to all tracks of the source device and the related target tracks.

EXTENTS

Specifies that the first and last track of a set of contiguous tracks (an extent pair) be part of this FlashCopy relationship. This set of tracks applies to both the source and target devices. The first track value must be lower or equal to the last track value in the track extent. All tracks within the track extent range become part of the FlashCopy relationship.

The following details the content of each track extent:

c1h1 Specifies the cylinder and head number of the beginning of the range. These numbers can be specified without the 'X' designation. For example, you would enter *cccchhhh* where the first four digits represent the beginning cylinder number and the second four digits represent the beginning head number.

c2h2 Specifies the cylinder and head number of the end of the range. These numbers can be specified without the 'X' designation. For example, you would enter *cccchhhh* where the first four digits represent the ending cylinder number and the second four digits represent the ending head number.

The delimiter separating each track specification (cccchhhh) can be a space or a comma (,). The range is verified to ensure that it is within the limits of the device. No intervening values can be omitted. **Example:** A full 3390-3 volume track extent is specified as *00000000 0D0A000E*.

Note:

1. In the FCESTABL command, the EXTENTS option specifies source volume begin and end track extents that are copied to the target volume begin and end track extents at the same location. This allows for a greater number of extents to be specified.
2. EXTENTS is a FlashCopy V1 function and is not supported when OPENDVCS(YES), REMOTE(YES), MODE(NOCOPY2COPY), MODE(ASYNC), or INCREMENTAL(YES) is specified.

Note:

XTNTLST

Specifies a list of extent sets that are in individual FlashCopy relationships. You can specify as many extent sets as allowed by the TSO command line limitation.

Note:

1. You can use XTNTLST only if FlashCopy V2 is enabled on the ESS.
2. You can use XTNTLST only if both the source and target volumes are 65,520 cylinders or smaller (not an EAV).
3. XTNTLST is not supported when OPENDVCS(YES), REMOTE(YES), INCREMENTAL(YES), or MODE(ASYNC) is specified.

The following details the information that each set of extents contains:

- The first and last track of a set of contiguous tracks on the source device
- The first and last track of a set of contiguous tracks on the target device

c1h1
Specifies the cylinder and head number of the beginning source track of the range. These numbers can be specified without the 'X' designation. For example, you would enter *cccchhhh* where the first four digits represent the beginning cylinder number and the second four digits represent the beginning head number.

c2h2
Specifies the cylinder and head number of the ending source track of the range. These numbers can be specified without the 'X' designation. For example, you would enter *cccchhhh* where the first four digits represent the ending cylinder number and the second four digits represent the ending head number.

c1h1
Specifies the cylinder and head number of the beginning target track of the range. These numbers can be specified without the 'X' designation. For example, you would enter *cccchhhh* where the first four digits represent the beginning cylinder number and the second four digits represent the beginning head number.

c2h2
Specifies the cylinder and head number of the ending target track of the range. These numbers can be specified without the 'X' designation. For example, you would enter *cccchhhh* where the first four digits represent the ending cylinder number and the second four digits represent the ending head number.

The beginning tracks (source and target) in each extent set must be equal to, or lower than, the ending tracks (source and target) in each extent set. The total number of source tracks in each extent set must equal the total number of target tracks in each extent set.

XXTNTLST

Specifies a list of extent sets that are in individual FlashCopy relationships. You can specify as many extent sets as allowed by the TSO command line limitation.

Note:

1. You can use XXTNTLST only if FlashCopy V2 is enabled on the storage control.
2. You must use XXTNTLST if either or both the source and target volumes are larger than 65,520 cylinders (EAV).
3. XXTNTLST is not supported when OPENDVCS(YES), REMOTE(YES), or INCREMENTAL(YES) is specified.

The following details the information that each set of extents contains:

- The first and last track of a set of contiguous tracks on the source device
- The first and last track of a set of contiguous tracks on the target device

c1h1 Specifies the cylinder and head number of the beginning source track of the range. These numbers can be specified without the 'X' designation. For example, you would enter *cccccccch* where the first seven digits represent the beginning cylinder number and the last digit represents the beginning head number.

c2h2 Specifies the cylinder and head number of the ending source track of the range. These numbers can be specified without the 'X' designation. For example, you would enter *cccccccch* where the first seven digits represent the ending cylinder number and the last digit represents the ending head number.

c1h1 Specifies the cylinder and head number of the beginning target track of the range. These numbers can be specified without the 'X' designation. For example, you would enter *cccccccch* where the first seven digits represent the beginning cylinder number and the last digit represents the beginning head number.

c2h2 Specifies the cylinder and head number of the ending target track of the range. These numbers can be specified without the 'X' designation. For example, you would enter *cccccccch* where the first seven digits represent the ending cylinder number and the last digit represents the ending head number.

The beginning tracks (source and target) in each extent set must be equal to, or lower than, the ending tracks (source and target) in each extent set. The total number of source tracks in each extent set must equal the total number of target tracks in each extent set.

ACTION

An optional parameter that specifies special action to take during the FlashCopy establish processing. Valid values are:

FREEZE

Specifies that the FlashCopy source volume is to be part of a FlashCopy consistency group. The FlashCopy relationship is to be established between the source and target volumes, or extents, and all I/O to the source volume will be held (results in a long busy) until one of the following conditions is met:

- A FlashCopy withdraw with action THAW is processed by the LSS where the volume resides
- A two-minute timer has expired

ACTION=FREEZE is mutually exclusive with MODE(ASYNC).

FRR

Fast Reverse Restore is a function to be used with Global Mirror or Metro/Global Mirror for ESS when recovering from an outage. This reverses the direction of the FlashCopy relationship, restoring the source volume to the state it was in when it last flashed to the target. Changed tracks are copied back from the target to the source.

Note:

1. If you use FRR, you must also use YES for TGTPRIM.
2. You cannot use ACTION with REMOTE(YES).

SETGTOK
Indicates whether the target of the specified full volume relationship can be a space efficient volume. If an out of space condition occurs, the specified action is taken. Valid values are:

NO indicates that the target can not be a space efficient volume. The default is NO.

YES
indicates that the target can be a space efficient volume. If an out of space condition occurs, the relationship is failed.

PRESERVEMIRROR or PRESMIR
Indicates the level of requirement for a Preserve Mirror operation when the specified target is a PPRC primary device. Valid values are:

NO indicates that the FCESTABL request is to be performed without considering a Preserve Mirror operation. This is the default.

PREFERRED or PREF
indicates that if the specified target device is a Metro Mirror primary device, it would be preferable for the pair to not go into a duplex pending state as the result of the FCESTABL request, but the request should be performed even if the pair does go into a duplex pending state.

PREFERRED is not supported in a Multi-Target Mirror configuration.

REQUIRED or REQ
indicates that if the specified target device is a Metro Mirror primary device, the pair must not go into a duplex pending state as the result of the FCESTABL request. If the duplex pending state cannot be avoided, the request fails.

Note:

1. If you use PREFFERED or REQUIRED, you must:
   - Also use TGTPPRIM(YES)
   - Not use REMOTE(YES) or ACTION(FRR).
2. If Preserve Mirror is used in combination with incremental FlashCopy, the use of PREFERRED or REQUIRED when taking an increment needs to be consistent with the existing relationship. So, if the existing incremental relationship was established without Preserve Mirror, an increment cannot be taken with Preserve Mirror required, because the remote relationship will not exist, meaning an increment cannot be taken at the remote site.

SUBCHSET
Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is supported by the host system's processor and configured for the device.

If you omit the parameter, the command is issued to the device that is logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

TSUBCHSET
Specifies which subchannel set is to be used to get information about the

target device specified with the TDEVN parameter. This is the subchannel set for the device as defined in the Hardware Configuration Dialog (HCD). The values are determined by what is supported by the host system's processor and configured for the device.

If you omit the parameter, the device that is logically in subchannel set 0 will be used. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

FCESTABL command examples

The following examples show how the FCESTABL command is used to establish FlashCopy relationships using various options.

FCESTABL example 1: This example establishes two FlashCopy relationships using the XTNTLST parameter.

```
FCESTABL SDEVN(042A) TDEVN(0160) MODE(NOCOPY)
  XTNTLST (01000004 0357000A 13000001 15570007, 02AC0006 02FF000C 18A00000 18F30006)
```

Figure 58. FCESTABL Example 1

This command establishes two FlashCopy relationships as follows:

- Between the source device extent ranges 01000004 through 0357000A and the offline target device extent ranges 13000001 through 15570007
- Between the source device extent ranges 02AC0006 through 02FF000C and the offline target device extent ranges 18A00000 though 18F30006.

No background copy is initiated.

Note: The target device must be offline because the ONLINTGT(YES) parameter was not used. The relationship cannot be established if the target device is online and the ONLINTGT parameter was not specified. 'Online' means not only online to the host from which you are operating, but potentially online to any other host attached to the ESS. If the device is in a path group to another host, it is considered 'online.'

FCESTABL example 2: This example shows how a FlashCopy relationship is established for CKD devices without a full-volume background copy.

```
FCESTABL SDEVN(042A) TDEVN(0160) MODE(NOCOPY) ONLINTGT(YES)
```

Figure 59. FCESTABL Example 2

This relationship can be established even if the target device is online because the ONLINTGT parameter is set to YES. Since MODE(NOCOPY) is specified, no background copy is initiated and the TARGET device will not be written until the SOURCE is updated by the application, at which time the T0 (pre-update) data is preserved on the target.

FCESTABL example 3: This example establishes a FlashCopy relationship between open (FB) devices using a CKD device, identified by the DEVN keyword, as an access device.

```
FCESTABL DEVN(0240) SOURCE(22940 0A 10) TARGET(22940 0C 18) OPENDVCS(YES)
```

Figure 60. FCESTABL Example 3

DEVN 0240 is a CKD device in the same storage subsystem cluster as the FB SOURCE volume.

FCESTABL example 4: This example establishes an inband FlashCopy relationship between CKD devices.

```
FCESTABL DEVN(042A) SOURCE(18432 12 08) TARGET(18432 14 10) MODE(COPY)
  SSID(1418) REMOTE(YES)
```

Figure 61. FCESTABL Example 4

The REMOTE(YES) parameter specifies that this relationship is to be located on the remote storage subsystem identified by SSID 1418. This must be the same as the SSID value specified for the PPRC secondary in the PPRC CESTPAIR command. DEVN 042A is the PPRC primary through which the FCESTABL command will be passed.

FCESTABL example 5: This example establishes an inband FlashCopy relationship between open (FB) devices.

```
FCESTABL DEVN(1282) SOURCE(26484 0D 20) TARGET(26484 18 2A)
  OPENDVCS(YES) REMOTE(YES)
```

Figure 62. FCESTABL Example 5

Since this request is for FB devices on a remote system, DEVN 1282 is a CKD device on the same storage subsystem cluster as the PPRC primary paired with the PPRC secondary specified as FlashCopy SOURCE.

FlashCopy query (FCQUERY) command

Use the FCQUERY command to determine the copy status of any device available to the host system. If you issue a command to a device not in an ESS, certain data will not necessarily be available. The information reflects the storage control type, and whatever information is available about the device will be returned.

When you issue a FCQUERY command to an ESS device that is part of a FlashCopy V1 relationship, the specified device information includes whether the device is the source or target of a FlashCopy relationship. The FCQUERY command for a device enabled with FlashCopy V2, provides the number of total relationships, source and/or target, that are active on the device. As much information about the specified device as is available is returned. Because status information is extremely volatile in a shared device environment, these conditions might not exist when the next FlashCopy command is issued.

FCQUERY command syntax

The syntax of the FCQUERY command is:

```
►►──FCQUERY──DEVN(ssss)──┬──────────────────────┬──────────────────────────────────────────────►
                         └─SUBCHSET(subchset)─┘

►──┬──┬─────────────────┬──┬──────────┬──────────────────────────────────────┬────────────────►◄
   │  └─┤ Open Devices ├┘  └─UNFORMAT─┘                                      │
   └─SHOWRELS──┬─(NO)─────────────────────────┬──┬──────────────────────────┬┘
               ├─(ALL)────────────────────────┤  └─STARTADDR──(track-address)─┘
               └─(number-of-relationships)────┘
```

Open Devices:

```
├──QRYDVC(dev)──┬──┬─────────────────┬──┬──────────────────────┬──────────────┬──────────────────┤
                │  └─OPENDVCS(YES)───┘  └─REMOTE──┬─(YES)─┬────┘              │
                │                                 └─(NO)──┘                   │
                └──┬─────────────────┬──┬─────────────┬──QRYSSID(ssid)────────┘
                   └─OPENDVCS(NO)────┘  └─REMOTE(YES)─┘
```

Required parameters

DEVN

When querying a local CKD device, this specifies the four-digit hexadecimal device number of the volume, *ssss*.

When querying a local fixed block device (OPENDVCS(YES) is specified), DEVN must specify a CKD access volume located in the same subsystem cluster as the FB device identified by QRYDVC in this command.

When querying a remote fixed block device (both OPENDVCS(YES) and REMOTE(YES) are specified), DEVN must specify a CKD access volume located in the same subsystem cluster as the PPRC primary device that is paired with the PPRC secondary acting as the FlashCopy source.

When querying a remote CKD device (REMOTE(YES) is specified), DEVN must specify the PPRC primary device that is paired with the PPRC secondary acting as the FlashCopy source.

ssss must be a 4-digit hexadecimal address of a configured device with a UCB on the System z system issuing the command.

Optional parameters

QRYSSID

When REMOTE(YES) is specified without OPENDVCS(YES), *ssid* is the 4-digit hexadecimal address of the storage subsystem where the FlashCopy request is to occur. This must be the same value as that specified for SSID of the secondary device on the PPRC establish pair command.

QRYDVC

When OPENDVCS(YES) or REMOTE(YES) is specified, the value for *dev* is:

volume's storage control serial number

This can include up to 10 digits, depending on the type of storage control.

logical storage subsystem

This is a two-digit hexadecimal value for the logical subsystem (LSS) for the device (ESS only).

Note: The LSS number is required if the storage control supports logical subsystems (like the ESS) and not allowed if the storage control does not support logical subsystems.

channel connection address or logical unit number
The channel connection address is a two-digit hexadecimal value. The logical unit number is the two-digit hexadecimal value for the open device.

OPENDVCS
Specifies whether Open System (fixed block) devices or CKD devices are addressed by this command. The values are:

NO The DEVN parameter identifies a CKD device. The default is NO.

YES
Specifies that the DEVN and QRYDVC parameters identify fixed block devices using a CKD access device.

REMOTE
Specifies whether the request is for a subsystem that is not directly attached to the issuing processor. The values are:

NO The request is directed to a device on a subsystem locally attached to the issuing processor. The default is NO.

YES
The request is directed to a remote subsystem using Inband subsystem functions.

When specifying an Inband request for a CKD secondary device, DEVN must identify a PPRC primary device in a subsystem accessible by the host processor issuing the request, and the device specified as the FlashCopy source must also be the PPRC secondary for the DEVN device.

When specifying an Inband request for an open (fixed block) secondary device, DEVN must identify an online CKD System z device located in the same subsystem cluster as the fixed block device of the PPRC pair with the secondary designated as FlashCopy target. The device specified as the FlashCopy Source device must be the PPRC secondary of a FB PPRC pair.

SHOWRELS
Indicates whether or not the device's relationship table information is to be formatted and displayed. The values for SHOWRELS are:

NO
indicates that the relationship table information for the device is not to be displayed. This is the default.

ALL
indicates that the relationship table information for all of the relationships for the device is to be formatted and displayed.

number-of-relationships
indicates that the relationship table information for the device is to be formatted and displayed up to *number-of-relationships*. Valid values are 1 through 9999.

This requires FlashCopy Version 2. If the device does not support FlashCopy Version 2, the FlashCopy Version 1 relationship table query results are displayed.

The relationship table information does not appear in the SYSLOG.

STARTADDR
Indicates the starting track for a query on which the device's relationship table information is to be displayed. The results start with relationships intersecting with *track-address*. *track-address* indicates the starting track by cylinder and head number.

UNFORMAT
Specifies that the information that is returned from the FCQUERY is to be formatted as a string of values separated by commas, similar to the information string returned to the ANTRQST API caller of FCQUERY. The default is formatted data.

Note: The UNFORMAT report is replaced by the FORMAT=FQMAP on the ANTRQST FCQUERY request. Thus, the UNFORMAT report is not updated.

SUBCHSET
Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is supported by the host system's processor and configured for the device.

If you omit the parameter, the command is issued to the device that is logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

FCQUERY command examples

The following examples demonstrates how to issue a FCQUERY command. See "Displaying information about FlashCopy volumes" on page 481 for output examples.

```
FCQUERY DEVN(0F4C)
```

Figure 63. FCQUERY Example 1: Query for a CKD Device

```
FCQUERY DEVN(042A) QRYDVC(28532 X'10' X'48') OPENDVCS(YES)
```

Figure 64. FCQUERY Example 2: Query for an Open Device

```
FCQUERY DEVN(9813) OPENDVCS(YES) REMOTE(YES) QRYDVC(26677 16 6F)
```

Figure 65. FCQUERY Example 3: Query for a Remote Open Device

```
FCQUERY DEVN(042A) QRYDVC(28440 X'0A' X'18') REMOTE(YES) QRYSSID(8700)
```

Figure 66. FCQUERY Example 4: Query for a Remote CKD Device

FlashCopy withdraw (FCWITHDR) command

The FCWITHDR command is used to end a FlashCopy relationship. The FCWITHDR command causes all or selected extents on a volume to be withdrawn.

This command can be used on FlashCopy relationships whether or not a background copy is in progress. The FCWITHDR command provides a method of manually withdrawing the relationship.

For Preserve Mirror relationships, the relationship is withdrawn immediately, at both the local and remote storage servers. See "Withdrawing a Preserve Mirror FlashCopy relationship" on page 491 for more information.

For FlashCopy V2, the FCWITHDR command allows specified (or all) target tracks on a source volume to be removed from their associated FlashCopy relationships. The XTNTLST parameter can specify a collection of extent sets to be withdrawn. The DDSW(YES) parameter can withdraw all source and target relationships that are active (for an entire volume or within the scope of an XTNTLST specification).

Note: When a FlashCopy NOCOPY relationship is ended, the track data on the target device is unpredictable and should not be used. If updates occur to source device tracks in the FlashCopy NOCOPY relationship, a copy of the source tracks from the point-in-time of the FlashCopy establish may or may not be written to the target device.

Tracks may be copied from the source to the target volume even if the source track is not changed. This includes the track that contains the volume label. Therefore, to avoid duplicate volume serial problems when the target device is later varied online, IBM recommends that you relabel the target volume after withdrawing a volume-level FlashCopy NOCOPY relationship.

FCWITHDR command syntax

The syntax of the FCWITHDR command is:

```
►►──FCWITHDR──┬─┬─SDEVN(ssss)─┤ Sub ├─┬─TDEVN(tttt)─┤ TSub ├─┬────────────┬─┬─┬──┤ Extents ├─┬─┬──►
              │ │                     │                       └─DDSW(NO)───┘ │ │               │ │
              │ │                     └─DDSW(YES)────────────────────────────┘ │               │ │
              │ └─TDEVN(tttt)──────────────────────────────────────────────────┘               │ │
              └─DEVN(devno)─┤ Sub ├─┤ Open Devices ├─┬────────────────┬────────────────────────┘ │
                                                     └─ACTION(THAW)───┘                          

►──┬──────────────────────────┬──┬──────────────────────┬──────────────────────────────────────►◄
   └─ACTION(─┬─COMMIT─┬─)─────┘  └─TGTUCB(─┬─YES─┬─)────┘
             └─REVERT─┘                    └─NO──┘
```

Sub (Subchannel Set):

```
├──┬──────────────────────┬──────────────────────────────────────────────────────────────────┤
   └─SUBCHSET(subchset)───┘
```

TSub (Target Subchannel Set):

```
├──┬────────────────────────┬────────────────────────────────────────────────────────────────┤
   └─TSUBCHSET(tsubchset)───┘
```

Extents:

```
├──┬──────────────────────────────────────────────┬──────────────────────────────────────────┤
   │              ┌─,──────────────────────┐      │
   │              ▼                        │      │
   ├─XTNTLST──(───c1h1──c2h2──c1h1──c2h2───┴──)───┤
   └─XXTNTLST──(──c1h1──c2h2──c1h1──c2h2──)───────┘
```

Open Devices:

```
├──SOURCE(srcid)──TARGET(tgtid)──┬─OPENDVCS(YES)──REMOTE(─┬─YES─┬─)──────┬───────────────────┤
                                 │                        └─NO──┘         │
                                 └─OPENDVCS(NO)──REMOTE(YES)──SSID─(ssid)─┘
```

Required parameters

SDEVN or TDEVN

Specifies the type of device to withdraw from a FlashCopy relationship. You must specify at least one of the parameters when withdrawing a local CKD from a FlashCopy relationship. The following values are associated with these required parameters:

SDEVN

Specifies a four-digit hexadecimal device number of a local CKD device used as the FlashCopy source.

Note:

1. If you specify the SDEVN parameter, you must also specify the TDEVN parameter, unless you also specify DDSW(YES). However, you can specify the TDEVN alone.

2. If you specify DDSW(YES), you must specify the SDEVN parameter, in which case the TDEVN parameter is ignored.

TDEVN
Specifies a four-digit hexadecimal device number of a local CKD device used as the FlashCopy target.

DEVN
When OPENDVCS(YES) is specified, DEVN identifies a CKD access volume located in the same subsystem cluster as the open volume specified as SOURCE in this command.

When OPENDVCS(YES) and REMOTE(YES) are both specified, DEVN identifies a CKD access volume located in the same subsystem cluster as the PPRC primary paired with the PPRC secondary acting as the FlashCopy source.

When REMOTE(YES) is specified for CKD volumes ((OPENDVCS(NO) is specified or allowed to default), DEVN identifies the PPRC primary device paired with the PPRC secondary acting as FlashCopy source.

DEVN must be a 4-digit hexadecimal address of a configured device with a UCB on the System z system issuing the command.

SOURCE
When OPENDVCS(YES) or REMOTE(YES) is specified, this identifies the source volume to be withdrawn from a FlashCopy relationship, by storage control serial number (*serialno)*, primary volume channel connection address (*CCA*), and logical storage subsystem (*lss*).

Where:

serialno
Specifies the storage control serial number that can include up to 10 digits, depending on the type of storage control.

lss
Specifies a 2-digit hexadecimal value that specifies the logical subsystem (LSS) for the device (ESS only).

Note: The LSS number is required if the storage control supports logical subsystems (like the ESS) and not allowed if the storage control does not support logical subsystems.

cca
Specifies the 2-digit channel connection address

lun
Specifies the 2-digit hexadecimal LUN address

TARGET
When OPENDVCS(YES) or REMOTE(YES) is specified, this identifies a target volume to withdraw from a FlashCopy relationship. The descriptions for these values are the same as those shown for the SOURCE parameter.

Optional parameters

OPENDVCS
Specifies whether Open System (fixed block) devices or CKD devices are addressed by this command. The values are:

NO The required parameters (SDEVN and TDEVN) identify CKD devices. The default is NO.

YES
Specifies that SOURCE and TARGET parameters identify Open System (fixed block) devices using a CKD access device.

REMOTE
Specifies whether the request is for a subsystem that is not directly attached to the issuing processor. The values are:

NO The request is directed to a device on a subsystem locally attached to the issuing processor. The default is NO.

YES
The request is directed to a remote subsystem using Inband subsystem functions. When REMOTE(YES) is specified, the request must be for a full volume.

When specifying an Inband request for a CKD secondary device, DEVN must identify a PPRC primary device in a subsystem accessible by the host processor issuing the request, and the device specified as the FlashCopy source must also be the PPRC secondary for the DEVN device.

When specifying an Inband request for an open (fixed block) secondary device, DEVN must identify an online CKD System z device located in the same subsystem cluster as the fixed block device serving as the primary of a PPRC pair where the secondary will be the FlashCopy source device.

Note: REMOTE(YES) is not valid when the EXTENTS or XTNTLST parameters are specified.

SSID
When REMOTE(YES) is specified without OPENDVCS(YES), this is the four-digit hexadecimal address identifying the storage subsystem where the remote FlashCopy is located. This must be the same value as that specified for SSID on the PPRC CESTPAIR command.

DDSW
Specifies whether you want the deleted data space withdraw (DDSW) function to be performed on the device that is specified with the SDEVN parameter. This parameter applies to devices that are enabled with FlashCopy V2.

One of the following values must be used with the DDSW parameter:

YES
Specifies to withdraw all eligible relationships on the device specified with the SDEVN parameter from their associated FlashCopy relationships. For additional information, see “Using DDSW(YES)” on page 492. DDSW(YES) is not valid when OPENDVCS(YES) or REMOTE(YES) is specified.

NO Specifies that withdraw processing be performed without deleted data space withdraw considerations. The default is NO. For additional information, see “FlashCopy V2 withdraw options” on page 491.

XTNTLST, or XXTNTLST
Apply to devices that are enabled with FlashCopy V2. The parameters are mutually exclusive, specifying that FlashCopy withdraw actions apply to the ranges of source and target extents that are specified within the list. The parameters are not valid when OPENDVCS(YES) or REMOTE(YES) is specified.

Each set of extents contains the following information:

- The first and last track of a set of contiguous tracks on the source device

- The first and last track of a set of contiguous tracks on the target device

XTNTLST
This parameter applies to devices that are enabled with FlashCopy V2. The parameter specifies that FlashCopy withdraw actions apply to the ranges of source and target extents that you specify within the list. XTNTLST is not valid when OPENDVCS(YES) or REMOTE(YES) is specified

Note: You can use XTNTLST only if both the source and target volumes are 65,520 cylinders or smaller (not an EAV).

c1h1 Specifies the cylinder and head number of the beginning source track of the range. These numbers can be specified without the 'X' designation. For example, you would enter *ccccchhhh* where the first four digits represent the beginning cylinder number and the second four digits represent the beginning head number.

c2h2 Specifies the cylinder and head number of the ending source track of the range. These numbers can be specified without the 'X' designation. For example, you would enter *ccccchhhh* where the first four digits represent the ending cylinder number and the second four digits represent the ending head number.

c1h1 Specifies the cylinder and head number of the beginning target track of the range. These numbers can be specified without the 'X' designation. For example, you would enter *ccccchhhh* where the first four digits represent the beginning cylinder number and the second four digits represent the beginning head number.

c2h2 Specifies the cylinder and head number of the ending target track of the range. These numbers can be specified without the 'X' designation. For example, you would enter *ccccchhhh* where the first four digits represent the ending cylinder number and the second four digits represent the ending head number.

The beginning tracks (source and target) in each extent set must be equal to, or lower than, the ending tracks (source and target) in each extent set. The total number of source tracks in each extent set must equal the total number of target tracks in each extent set.

If you do not specify the XTNTLST parameter, the FlashCopy FCWITHDR command defaults to processing all tracks of the specified device(s). For additional information about using this parameter, see "Limiting the scope of a withdraw using the XTNTLST" on page 493.

XXTNTLST

Note: You must use XXTNTLST if either or both the source and target volumes are >65,520 cylinders (EAV).

c1h1 Specifies the cylinder and head number of the beginning source track of the range. These numbers can be specified without the 'X' designation. For example, you would enter *cccccccch* where the first seven digits represent the beginning cylinder number and the last digit represents the beginning head number.

c2h2 Specifies the cylinder and head number of the ending source track of the range. These numbers can be specified without the 'X' designation. For example, you would enter *cccccccch* where the first seven digits represent the ending cylinder number and the last digit represents the ending head number.

c1h1 Specifies the cylinder and head number of the beginning target track of the range. These numbers can be specified without the 'X' designation. For example, you would enter *cccccccch* where the first seven digits represent the beginning cylinder number and the last digit represents the beginning head number.

c2h2 Specifies the cylinder and head number of the ending target track of the range. These numbers can be specified without the 'X' designation. For example, you would enter *cccccccch* where the first seven digits represent the ending cylinder number and the last digit represents the ending head number.

The beginning tracks (source and target) in each extent set must be equal to, or lower than, the ending tracks (source and target) in each extent set. The total number of source tracks in each extent set must equal the total number of target tracks in each extent set.

ACTION
An optional parameter that specifies special action to take during the FlashCopy withdraw processing. Valid values are:

THAW
Specifies that FlashCopy consistency group formation is complete and application I/O can now take place on all volumes in the LSS where the previous FlashCopy Establish request was issued with the ACTION FREEZE parameter.This option can only be specified with the DEVN parameter, not with SDEVN or TDEVN.

COMMIT
This specifies that the last consistency group created by the Global Mirror session is committed to the current state, and reverting to the previous consistency group state is no longer possible. See “Moving a Global Mirror session to the recovery site in an unplanned outage” on page 446 for guidelines on the use of this option.

REVERT
Revert is a function to be used with Global Mirror or Metro/Global Mirror for ESS when recovering from an outage. It specifies a rollback to the state saved by a previous automatic FlashCopy establish command. The FlashCopy relationship is not removed (withdrawn), the FlashCopy target is rolled back to the previous consistency group created by the Global Mirror session. See “Moving a Global Mirror session to the recovery site in an unplanned outage” on page 446 for guidelines on the use of this option.

TGTUCB(YES|NO)
Indicates whether an MVS device number is being used for target addressability.

YES
Indicates that the SDEVN and TDEVN keywords are being used to identify the source and target devices in the relationship using MVS device numbers. The default is YES.

NO Indicates that an MVS device number will not be used for the target device (TDEVN keyword will not be used). Instead, the SDEVN and TARGET keyword will be used.

Specifying TGTUCB(NO) will be ignored when specified with REMOTE(YES) or *OPENDVCS(YES)*. This is because these two types of requests already use the TARGET keyword instead of the TDEVN keyword and the system does not have direct addressability to the target volume in the relationship.

SUBCHSET
Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is supported by the host system's processor and configured for the device.

If you omit the parameter, the command is issued to the device that is logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

TSUBCHSET
Specifies which subchannel set is to be used to get information about the target device specified with the TDEVN parameter. This is the subchannel set for the device as defined in the Hardware Configuration Dialog (HCD). The values are determined by what is supported by the host system's processor and configured for the device.

If you omit the parameter, the device that is logically in subchannel set 0 will be used. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

FCWITHDR command examples

The following are examples of using the FCWITHDR command:

FCWITHDR example 1:

```
FCWITHDR SDEVN(042A) TDEVN(0160)
  XTNTLST(00000000 0100000E 20000000 2100000E)
```

Figure 67. FCWITHDR Command Example 1

This command withdraws the FlashCopy relationship between the source and target extents specified with XTNTLST parameter.

FCWITHDR example 2:

```
FCWITHDR SDEVN(042A) DDSW(YES)
```

Figure 68. FCWITHDR Command Example 2

This command withdraws all source extents on the source device 042A from their associated FlashCopy relationships regardless of the location of their corresponding target extents, and all target extents on the source device regardless of the location of the corresponding source extents.

FCWITHDR example 3:

```
FCWITHDR DEVN(4181)  OPENDVCS(YES) SOURCE(26677 16 6F) TARGET(26677 11 23)
```

Figure 69. FCWITHDR Command Example 3

This command withdraws the identified open (FB) source and target volumes from their FlashCopy relationship.

FCWITHDR example 4:

```
FCWITHDR DEVN(4181) SOURCE(26677 0A 2D) TARGET(26677 03 91)
```

Figure 70. FCWITHDR Command Example 4

Example 4 withdraws the identified CKD source and target volumes from their FlashCopy relationship.

FCWITHDR example 5:

```
FCWITHDR DEVN(1282) SOURCE(26484 16 20) TARGET(26484 18 2A)
  OPENDVCS(YES) REMOTE(YES)
```

Figure 71. FCWITHDR Example 5

Since this request is for FB devices on a remote system, DEVN 1282 is a CKD device on the same storage subsystem cluster as the PPRC primary paired with the PPRC secondary specified as FlashCopy SOURCE.

Part 6. SnapShot copy

This provides the information to help you use the SnapShot copy function.

Chapter 27. What is SnapShot copy?

This section explains what the SnapShot copy function is and how it works, and describes the various hardware and software products that SnapShot copy requires. SnapShot is only supported on RVA devices.

SnapShot copy is an Advanced Copy Services function that can quickly move data from the source device to the target device. DFSMSdss calls SnapShot copy when you request the function for data on hardware that supports the SnapShot capability. The two types of SnapShot copy includes:

Regular SnapShot
: Data is "snapped" (quickly copied) directly from the source location to the target location. This function occurs when you issue a DFSMSdss COPY command to copy volumes, tracks, or data sets from one DASD volume to another. DFSMSdss uses this method whenever the source and target data are located on like devices in the same partition on the same RVA subsystem, and no reblocking is required. DFSMSdss can use this method whether or not the CONCURRENT keyword is specified. With "regular" SnapShot, the copy of the data is logically and physically complete as soon as the snap is complete.

Virtual-concurrent-copy SnapShot
: Data is snapped from the source location to an intermediate location and then gradually copied to the target location using normal I/O methods. This function occurs when you issue either the DFSMSdss COPY or DUMP command and you specify the CONCURRENT keyword. As the name implies, this method operates almost identically to existing concurrent copy (CC) support.

  All DFSMSdss users and callers of the DFSMSdss API (such as DFSMShsm, DB2, and IMS) can continue to use the CONCURRENT keyword and receive functionally identical CC support. Using virtual-concurrent-copy SnapShot, the copy or dump of the data is logically complete after the source data is snapped to the intermediate location, and then physically complete after the data has been moved to the target media. It is also possible to perform concurrent copy on VM minivolumes using virtual-concurrent-copy SnapShot. VM minivolumes are supported if you are using RVA devices to the extent that these devices are supported by IBM Extended Facilities Product (IXFP) device reporting.

In this topic

The following sections are included in this topic:

SnapShot copy requirements

There are both software and hardware prerequisites for SnapShot copy operation.

Determining SnapShot copy software requirements

DFSMS/MVS Version 1 Release 3 and subsequent releases provide SnapShot copy support. You must also have the SnapShot 1.2 and IXFP 2.1 program products installed. The SIBFVP and SIBLLAPI load modules must be in a library in the LINKLST concatenation.

Determining SnapShot copy hardware requirements

MVS/ESA SP 4.3 runs on the following processors:

- All ES/9000 models that support ESA/390
- All ES/3090-9000T models that support ESA/370 or ESA/390
- All ES/3090 E, S, J, and JH models at the SEC level for ESA/370

All RVA storage subsystems support SnapShot copy. SnapShot copy operates with both 3380 and 3390 devices, but does not operate with 3390 devices in 3380 track-compatible mode.

Making cache available

To use SnapShot copy with a particular cached storage subsystem, the subsystem cache must be active. The statuses of volume-level caching, nonvolatile storage (NVS), DASD fast write, and dual copy have no effect on whether or not you can use SnapShot copy. Their statuses, however, can affect the overall performance of volumes that have active SnapShot sessions.

Using SnapShot copy

This section describes the SnapShot copy operation and explains how the hardware and software components work together.

Steps for initializing sessions

Functionally, the data that you are dumping or copying is unavailable only long enough for DFSMSdss to initialize a SnapShot copy session. Initialization takes only a matter of seconds to complete successfully. The session initialization process is as follows:

1. Determine an appropriate time to start a backup of databases, and stop update activity to those databases.
2. Use DFSMSdss to perform a copy or dump of the data with SnapShot copy. You can use DFSMSdss directly from JCL or from an application through the DFSMSdss application interface.

Normal application activity resumes as soon as the SnapShot copy session initialization is complete.

Steps for providing a SnapShot working-space data space

Perform the following steps to provide a SnapShot working-space data space:

1. Ensure that adequate space is available before using the CONCURRENT keyword on DFSMSdss commands that refer to data sets on an RVA subsystem. The virtual-concurrent-copy SnapShot function requires some dedicated space on one or more volumes in the same RVA subsystem as the source data set.

Provide the necessary space by allocating dedicated data sets, called working-space data sets, with the following naming convention:

SYS1.ANTMAIN.S*sysname*.SNAP*nnnn*

Variable *sysname* is the system identifier, and *nnnn* is a four-digit decimal number in the value range 0001–9999. If the system identifier is eight characters, 'S' replaces the first character.

2. Ensure that you catalog the working-space data sets. DFSMSdss performs a numerically-sequential catalog search for each data set, starting with SYS1.ANTMAIN.S*sysname*.SNAP0001, until it encounters a catalog locate error. A catalog locate error indicates that DFSMSdss could not find the data set. DFSMSdss cannot use data sets as working space that are beyond the data set that DFSMSdss was unable to find.
3. Allocate the working-space data sets as physical sequential, nonextended-format data sets. The data sets can be SMS-managed or non-SMS-managed. If the catalog search for a working-space data set indicates that the data set is multivolume, DFSMSdss will not use it as a working-space data set.
4. Extend the data set by filling it with data before starting the DFSMSdss job if you want to allocate secondary space. The system data mover (SDM) does not extend a working-space data set. SDM holds an enqueue for the data set while the working space is being used during a SnapShot copy operation. It then releases the enqueue after all usage of the data set is completed. You can reallocate or extend a working-space data set when SDM does not have the data set enqueued. SDM uses the new size of the data set. You can add additional working-space data sets after ANTMAIN has completed the initialization process.
5. SDM uses these data sets the first time an out-of-working-space condition is encountered during a SnapShot copy operation. When this condition occurs, SDM refreshes the list of working-space data sets by performing a catalog search that starts with SYS1.ANTMAIN.S*sysname*.SNAP0001.

   The LRECL and block size can be any valid combination. The system uses the tracks within the data set as the target of SnapShot copy operations. Therefore, do not try to access these tracks with normal data access methods.

   Recommendation: It is recommended that you protect the working-space data sets by security, such as RACF, to ensure that unauthorized users do not have access to sensitive data.
6. Allocate data sets on a volume in each RVA subsystem that you have set aside for virtual-concurrent-copy SnapShot. If the RVA subsystem has more than one defined device type, you must allocate a working-space data set on each device type that contains a data set that DFSMSdss SnapShot copy will process. Device types are, for example, 3390 and 3380. Different models within each device type do not require separate working-space data sets.

   More than one system can simultaneously access data on the same RVA subsystem. If each system has DFSMSdss jobs that specify CONCURRENT, then each system requires at least one working-space data set.

   Example: DFSMSdss must allocate three working-space data sets to process data on an RVA subsystem from three MVS systems. The working-space data sets must include at least one of each device type that contains data that SnapShot will process.

   If there is insufficient working space, the SnapShot initialization for one or more data sets in the job fails. To avoid this, determine the largest total amount of data that you plan to process in a single DFSMSdss COPY or DUMP command on that RVA. The total size of all working-space data sets that you have allocated on each RVA subsystem should equal or exceed the determined amount of space.

Note: Do not place any working-space data set on a PPRC primary volume.

Copying data with SnapShot copy

When the initialization is complete, DFSMSdss automatically employs the SnapShot copy function for any data movement operation where the CONCURRENT option applies.

Moving data sets with SnapShot copy

DFSMSdss prefers to use SnapShot copy to quickly move the data from the source device to the target device. DFSMSdss can do this when the source and target devices are both in the RVA, and DFSMSdss does not need to manipulate data. This process, called regular SnapShot support, is much faster than traditional methods, especially when moving large amounts of data.

To use regular SnapShot, the following requirements must be met:

- Source and target device types (3390 or 3380, for example) must be the same.
- Source and target devices must be in the same RVA.
- Data must not require manipulation, such as reblocking.
- The data movement must not require utilities.

If the source data is in an RVA, DFSMSdss attempts to allocate the target data set on the same device type in the same RVA. This increases the probability that SnapShot can copy the data. DFSMSdss cannot allocate the target to use SnapShot if the source data set is multivolume, but is not entirely within one partition of an RVA subsystem. DFSMSdss allocates these data sets to whatever volumes are available, irrespective of their SnapShot capability.

DFSMSdss automatically performs SnapShot processing if possible, even when you specify the CONCURRENT keyword.

Determining how DFSMSdss determines volume eligibility for fast replication operations

Fast replication is a point-in-time function that allows data to be copied from a source location to a target location, such as FlashCopy on ESS devices and SnapShot on RAMAC Virtual Array (RVA) storage subsystems.

A QFRVOLS query of the ANTRQST API displays information regarding whether the source and target volumes are capable and eligible for fast replication operations: volume FlashCopy on the ESS or SnapShot on the RVA. If a volume can be used as a fast replication target volume, it can also be used as a fast replication source volume. Any condition that prevents a volume from being a fast replication target volume also prevents the volume from being a fast replication source volume.

The QFRVOLS query returns the type of fast replication for which each volume in a user-supplied volume list is eligible. If both the source and target volumes are eligible for participating in a fast replication operation, the extent sets that are included in the operation are copied. If a fast replication operation cannot be performed, the QFRVOLS query provides information that indicates why the volumes are not eligible for fast replication.

Using SnapShot copy, PPRC, and RVA together

The RAMAC Virtual Array (RVA) storage subsystem provides the unique ability to combine SnapShot copy functions with PPRC functions. For instance, you can use PPRC to migrate data to a different RVA subsystem; you can end the pairs and use SnapShot copy at the secondary subsystem. This ability can be very beneficial for data mining applications. However, there are some operational considerations regarding the interaction of SnapShot copy with volumes that are part of a PPRC pair.

- You cannot snap (copy) data onto any volume that is part of a PPRC pair. A primary volume cannot be the target of a SnapShot copy command even if it is in a suspended state.
- The primary device can be the source for a snap without having to change the state of the primary volume.
- SnapShot copy requires that space be available on the same RVA subsystem where the source of a snap resides. If you plan to use a SnapShot copy on an RVA subsystem that already has PPRC pairs, then provide volumes on that subsystem that are not part of a PPRC pair. You cannot, therefore, assign all of the volumes in a subsystem to be part of PPRC pairs and still use SnapShot copy on that subsystem.
- If you plan to use SnapShot copy at the secondary RVA subsystem, ensure that space is available at that subsystem. This real space must be independent of PPRC pairs. You cannot, therefore, have all 256 volumes as part of PPRC pairs.
- Do not put PPRC volumes into a SnapShot target storage group. Also, do not include a volume that is part of a PPRC pair within a SnapShot volume list. SnapShot does not check PPRC status. The SnapShot copy operation will fail if you attempt to allocate any part of a data set to a volume that is part of a PPRC pair. SnapShot copy uses all volumes available to it on a space-available basis when it dynamically allocates space for the target copy.
- You cannot perform any read or write I/O to a secondary volume of a PPRC pair. However, you can snap copies of data from the secondary device and bring the PPRC pairs back to active status. The following procedure outlines a quick way to do this:
  1. Suspend the pair.
  2. Recover the secondary, returning it to simplex.
  3. Snap the secondary.
  4. Issue a CESTPAIR command with the RESYNC parameter to copy only the changes.

  You can use a similar procedure in conjunction with the CGROUP command (MVS TSO command only) to snap data from secondary devices that contain related information.
  1. Issue a CGROUP command with FREEZE to suspend all pairs on the path between a primary logical storage subsystem and a secondary logical storage subsystem.
  2. Recover the secondary volumes, returning them to simplex.
  3. Snap the secondary volumes.
  4. Issue a CESTPAIR command with the RESYNC parameter to copy only the changed tracks of each volume.

SnapShot SMF information

The SDM writes an SMF type 42 subtype 4 record that contains session statistics. Among other things, you can use the information in this record to determine the following:

- Session initialization time
- Maximum size of host and storage subsystem sidefiles
- Number of intercepted writes.

For additional information about the SMF type 42 subtype 4 record, refer to "Concurrent copy information in SMF type 42 records" on page 569.

Part 7. Concurrent copy

This provides the information to help you use the concurrent copy function.

Chapter 28. What is concurrent copy?

This section explains what concurrent copy is and how it works, and describes the hardware and software requirements necessary to use concurrent copy.

Concurrent copy is a storage subsystem extended function that allows you to generate a copy or a dump of data while applications are updating that data. Other tools are available that allow you to dump databases while applications continue to update the databases. Concurrent copy, however, provides a copy of the data, in a consistent form, as it existed before the updates took place. Concurrent copy relies on cached storage subsystem Licensed Internal Code (LIC) and enhanced components of z/OS.

In this topic

The following sections are included in this topic:

Understanding the concurrent copy solution

Concurrent copy provides point-in-time data consistency, which is not possible with other online data dump techniques. The system serializes access to the data that you are dumping or copying just long enough for the concurrent copy session to initialize. This serialization takes a matter of seconds. Compare this to the quiesce and backup technique, which makes data unavailable for the entire duration of the dump (possibly hours).

The copy is *logically* complete as soon as you have initialized the concurrent copy environment. At that point, concurrent copy protects the original state of the data. After logical completion, the data is once again available for unrestricted application access. The copy is physically complete once the concurrent copy process finishes copying the data to the output device.

You can use concurrent copy to back up any data that can be backed up using DFSMSdss (or DFDSS) because DFSMSdss is the external interface to concurrent copy.

For example, you can back up IMS databases by using DFSMSdss. During the recovery process, IMS database recovery control (DBRC) coordinates recovery of the DFSMSdss dump and the application of updates from the IMS log. Conversely, DB2 (before Version 3) does not support those activities. Consequently, do not use concurrent copy with DB2 releases earlier than Version 3.

Concurrent copy requirements

This section describes the necessary software and hardware prerequisites for using concurrent copy.

Determining concurrent copy software requirements

All supported DFSMS/MVS and z/OS releases provide concurrent copy support. Concurrent copy requires the following components:

- DFSMSdfp for the system data mover (SDM), new management and storage class attributes, and ISMF support of those attributes
- DFSMSdss for the external interface to concurrent copy
- DFSMShsm to call concurrent copy during backup operations.

SMS is not necessary in order to use concurrent copy. You can use concurrent copy directly from DFSMSdss without SMS. However, DFSMShsm supports concurrent copy during automatic and aggregate backup only when SMS is active.

Concurrent copy with shared storage subsystems

In configurations where a single cached storage subsystem is attached to multiple host systems (running either in separate processors or separate logical partitions), any DFSMS/MVS or z/OS system can exploit concurrent copy. Although systems that are not running DFSMS/MVS or z/OS cannot initiate concurrent copy operations, such systems can share a cached storage subsystem (including volumes with active concurrent copy sessions) with DFSMS/MVS or z/OS systems provided that the non-DFSMS/MVS or z/OS systems support 32-byte sense information.

Determining concurrent copy hardware requirements

All cached storage subsystems provide support for concurrent copy.

Concurrent copy operates with both 3380 and 3390 devices, but does not operate with 3390 devices in 3380 track compatibility mode.

To use concurrent copy with a particular cached storage subsystem, the subsystem cache must be active. The statuses of cache, nonvolatile storage (NVS), DASD fast write, and dual copy have no effect on your use of concurrent copy. Their statuses, however, can affect the overall performance of volumes with active concurrent copy sessions.

Note: Be aware that if you disable cache on a storage subsystem that has active concurrent copy sessions, the cached storage subsystem will end the concurrent copy sessions. The operations that were using the sessions will also fail.

Determining concurrent copy resource needs

A concurrent copy operation requires the system resources that are described in this section.

Central and expanded storage

The system data mover (SDM) uses central storage for the host sidefiles that it maintains for concurrent copy sessions. A sidefile is a storage area that is used to maintain copies of tracks within a concurrent copy domain. A concurrent copy operation maintains a sidefile in storage subsystem cache and another in processor storage. The MVS real storage manager may page sections of the sidefile into expanded storage in response to varying system loads.

The size of the host sidefile is a secondary consideration when using concurrent copy. The SDM uses the sidefile only to store temporary copies of tracks as part of the intercepting write process. The size and use of the sidefile relate directly to the number of intercepted writes. The size of the host sidefile is unlikely to be a significant factor unless you use a concurrent copy operation during a period of very high update activity or in a very storage-constrained environment.

The SDM creates one host sidefile for each concurrent copy session. The host sidefiles for different concurrent copy sessions share a single data space, which can grow to a maximum size of 2 GB. The actual size of the data space is dynamic, however, and changes in response to the SDM requirements. The size depends on the following considerations:

- The number of active concurrent copy sessions
- The level of update activity
- The rate at which DFSMSdss reads data and writes it to the output media

In the unlikely event that the sidefiles will fill the data space, the SDM creates additional data spaces as necessary.

Another factor that affects the host sidefile size is the distribution of application writes across a device. I/O operations on a device typically concentrate on a set of tracks rather than uniformly distributing updates over the device. An application that updates a track causes an intercepted write. Subsequent updates to the same track do not result in an intercepted write. Similarly, as soon as DFSMSdss has processed a particular track, updates to that track do not result in an intercepted write. Concurrent copy only uses the host sidefile when there are intercepted writes. Therefore, the grouping of I/Os by applications significantly reduces the potential maximum size of the host sidefile.

Running many concurrent copy jobs simultaneously can cause auxiliary (AUX) storage shortages. You can control the total amount of AUX storage that is used by concurrent copy jobs at any given time. The MVS System Resources Manager (SRM) uses two different percentages (a lower value and an upper value) to determine when AUX storage shortages have been reached. SDM provides the ability to modify the concurrent copy AUX values that are used in evaluating the AUX storage that is used by concurrent copy. During concurrent copy processing, SDM subtracts these values from the two MVS percentages to determine if the total current system AUX storage percentage being used by the system is above these newly computed SDM percentages.

You can use the MODIFY MVS system command to change either the new or the existing concurrent copy job AUX delta values, as shown in "Example: specifying concurrent copy job AUX deltas."

Example: specifying concurrent copy job AUX deltas

1. Use the following command to modify the concurrent copy AUX delta to determine the point at which SDM will reject requests to start new concurrent copy tasks:

   ```
   F ANTMAIN,P .CMTUN+3A X'FFFF' X'nnnn'
   ```

   where *nnnn* is the new AUX delta value, in hexadecimal notation.

SDM subtracts this value from the lower MVS AUX storage limit percentage. The result is then compared to the percentage of AUX storage currently in use by the system. The default value is X'FFFF', indicating that SDM will not perform this check.

2. Use the following command to modify the concurrent copy AUX delta to determine the point at which SDM will terminate existing concurrent copy tasks:

   ```
   F ANTMAIN,P .CMTUN+3C X'0000' X'nnnn'
   ```

   where *nnnn* is the new AUX delta value, in hexadecimal notation.

   SDM subtracts this value from the upper MVS AUX storage limit percentage. The result is then compared to the percentage of AUX storage currently in use by the system. The default value is 0, indicating that SDM will terminate concurrent copy sessions when the upper limit is reached. A value of X'FFFF' indicates that SDM will not perform this check.

Storage subsystem cache

Concurrent copy, by design, minimizes the amount of cache storage that is used for sidefiles. Under normal circumstances, a concurrent copy session uses less than 0.5 MB of cache. It is very unlikely that using a few simultaneous concurrent copy sessions will significantly affect the hit ratios that you achieve with a cached storage subsystem. It is, however, possible that the storage used by a concurrent copy session could have an effect on hit ratios. This is especially likely in cases where the amount of cache is only barely adequate for normal requirements. In those cases, you should increase the size of the cache regardless of whether or not you use concurrent copy.

The following are several tactics that concurrent copy uses to minimize the amount of utilized cache:

- When reading data directly from DASD, the SDM and DFSMSdss inhibit the loading of tracks into the cache.
- When the cache sidefile holds no tracks, it occupies no cache space. A sidefile occupies cache storage only when it contains copies of tracks that are saved by the cached storage subsystem during processing of intercepted writes.
- The SDM reads and deletes tracks from a cache sidefile as soon as it detects that they are present.

The cached storage subsystem monitors the size of the cache sidefiles that an active concurrent copy sessions uses. In the highly unusual case of a system failure or software problem preventing the SDM from reading tracks, the storage subsystem cancels active concurrent copy sessions when the sidefiles occupy more than half of the cache. The storage subsystem cancels sessions by starting with the session with the largest cache sidefile. This action preserves cache storage for use by sharing systems, but does not affect data integrity. Data is safe because the sidefiles contain only images of tracks before an update was applied. When concurrent copy is active, the subsystem maintains the same level of integrity for update operations that it maintains when concurrent copy is not active.

Channel paths

Because concurrent copy operations generate additional I/Os to the volumes involved, the use of concurrent copy always increases channel utilization. The impact of using concurrent copy can vary, and depends on the following conditions:

- The number of simultaneous concurrent copy sessions.

- The DFSMSdss OPTIMIZE keyword.
- The number of paths available.

Using concurrent copy

This section describes concurrent copy operation and explains how the hardware and software components work together.

Steps for initializing sessions

Functionally, data being dumped or copied is unavailable only for the time it takes for DFSMSdss to start a concurrent copy session. Initialization takes only a matter of seconds to complete successfully. The session initialization process is a series of distinct steps, as follows:

1. Determine an appropriate time to start a backup of databases, and stop update activity to those databases.
2. DFSMSdss uses concurrent copy to perform a copy or dump of the data. You can use DFSMSdss directly from JCL or from an application through the DFSMSdss application interface.
3. DFSMSdss locates the data sets to process and breaks them down into a hierarchy of volumes and extents.
4. DFSMSdss interfaces with the system data mover to initialize a concurrent copy session.
5. The SDM determines which storage controls are attached to the affected volumes and defines a concurrent copy session to each storage control. The SDM also tells each storage control the volume and the extents on each volume that are included in this concurrent copy session. Once the SDM has completely initialized the session, it returns control to DFSMSdss.
6. DFSMSdss checks all SDM return codes. DFSMSdss releases serialization on those data sets or volumes that were successfully included in the concurrent copy session. DFSMSdss keeps serialization on data sets or volumes that are not included in the session.
7. DFSMSdss issues messages that indicate the status of the session. DFSMSdss uses the user interaction module (UIM) to update the calling program with the status of the session, if the application interface called DFSMSdss.
8. The application can take appropriate action, which may include restarting applications.

Copying the data

When the initialization is complete, DFSMSdss releases serialization and calls the UIM, as described in the previous section. DFSMSdss then starts to read the data to perform the dump or copy operation. Applications can resume processing the data while concurrent copy is making the copy. Meanwhile, the cached storage subsystems monitor the I/Os for updates to any tracks that are included in the concurrent copy session.

DFSMSdss uses the SDM to read data from the devices within the concurrent copy session. For data that was not included in the concurrent copy session, DFSMSdss reads the data itself in the same way as for a nonconcurrent copy operation. As DFSMSdss receives tracks from the SDM or reads them itself, it writes the data to the output device in the same way as for any DFSMSdss operation.

SMF information

The SDM writes an SMF type 42 subtype 4 record that contains session statistics. Among other things, you can use the information in this record to determine the following:

- Session initialization time
- Maximum size of host and cached storage subsystem sidefiles
- Number of intercepted writes

For additional information about the SMF type 42 subtype 4 record, refer to "Concurrent copy information in SMF type 42 records" on page 569.

Scheduling concurrent copy operations

Many factors influence when you can schedule concurrent copy operations.

Example: The structure of your overnight batch processing determines at what stages in the processing you can use concurrent copy. Similarly, availability of tape drives may restrict the intervals during which running a concurrent copy operation is feasible.

If other considerations are not a factor, use concurrent copy at times of lowest activity, especially lowest update activity. It may, however, be beneficial to use concurrent copy to back up data even during periods of higher I/O activity. In some cases, ensuring data availability may be more important than preserving levels of application performance.

Virtual-concurrent-copy FlashCopy function

Understanding the virtual-concurrent-copy

Virtual-concurrent-copy (VCC) FlashCopy is used on devices that have the FlashCopy function in place of the concurrent copy function. In the virtual-concurrent-copy FlashCopy function, the data is flash-copied from the source location to an intermediate location and then gradually copied to the target location using normal I/O methods. As the name implies, this method operates almost identically to existing concurrent copy (CC) support.

Allocating working-space dataset (WSDS)

To use the virtual-concurrent-copy function, allocate the working-space dataset. Working-space data sets serve as the mechanism for identifying where the temporary FlashCopy data is stored. Valid working-space dataset access must be through full track operations. The working-space data space allocation for virtual-concurrent-copy allows VSAM data sets.

Perform the following steps to provide a FlashCopy working-space data space:

1. Ensure that the FlashCopy relationship is active during the virtual-concurrent-copy, and that the target tracks are not updated.
2. Ensure that adequate space is available before using the CONCURRENT keyword on DFSMSdss commands that refer to data sets on an RVA Subsystem with FlashCopy capable subsystem. The virtual-concurrent-copy FlashCopy function requires some dedicated space on one or more volumes in the same RVA Subsystem with FlashCopy capable subsystem as the source data set.

Provide the necessary space by allocating dedicated data sets, called working-space data sets, with the following naming conventions:

- For sequential data sets, use

  HLQ.ANTMAIN.FCWK*nnnn* v

- For VSAM data sets, use

  HLQ.ANTMAIN.FCWK*nnnn*.DATA

Variable *HLQ* is the high level qualifier specified in PARMLIB and *nnnn* is a four-digit decimal number in the value range 0001–9999.

3. Ensure that you catalog the working-space data sets.
4. Allocate the working-space data sets as physical sequential, nonextended-format data sets. The data sets can be SMS-managed or non-SMS-managed. If you use the virtual-concurrent-copy on VSAM data sets, allocate the working-space data sets as non-indexed data sets, such as LDS or ESDS.
5. Extend the data set by filling it with data before starting the DFSMSdss job if you want to allocate secondary space. The system data mover (SDM) does not extend a working-space data set. SDM holds an enqueue for the data set while the working space is being used during a FlashCopy operation. It then releases the enqueue after all usage of the data set is completed. You can reallocate or extend a working-space data set when SDM does not have the data set enqueued. SDM uses the new size of the data set. You can add additional working-space data sets after ANTMAIN has completed the initialization process.
6. SDM uses these data sets the first time an out-of-working-space condition is encountered during a FlashCopy operation. When this condition occurs, SDM refreshes the list of working-space data sets by performing a catalog search.

   The LRECL and block size can be any valid combination. Do not try to access these tracks with normal data access methods.

   Recommendation: It is recommended that you protect the working-space data sets by security, such as RACF, to ensure that unauthorized users do not have access to sensitive data.
7. Allocate data sets on a volume in each FlashCopy capable subsystem that you have set aside for virtual-concurrent-copy FlashCopy. If the RVA Subsystem with FlashCopy capable subsystem has more than one defined device type, you must allocate a working-space data set on each device type that contains a data set that DFSMSdss FlashCopy copy will process. Device types are, for example, 3390 and 3380. Different models within each device type do not require separate working-space data sets.

   More than one system can simultaneously access data on the same RVA Subsystem with FlashCopy capable subsystem. The working-space data set is enqueued while the DFSMSdss job is using it. Other jobs on the same system can share the working-space data set provided enough space is available, but other systems can not. If multiple systems have DFSMSdss jobs which will use working-space data sets concurrently, then sufficient working -space data sets must be allocated to support the number of systems concurrently running jobs.

   If there is insufficient working space, the FlashCopy initialization for one or more data sets in the job fails. To avoid this, determine the largest total amount of data that you plan to process in a single DFSMSdss COPY or DUMP command on that RVA Subsystem with FlashCopy capable subsystem. The total size of all working-space data sets that you have allocated on each FlashCopy capable subsystem should equal or exceed the determined amount of space. You might also specify EF to get the larger allocation limits.

8. Issue system command F ANTMAIN,REFRESHWS on all systems that will run Virtual Concurrent Copy jobs. This is necessary to ensure that Virtual Concurrent Copy is attempted first when the PREFERRED, ANYPREFERRED, REQUIRED, or ANYREQUIRED option is in effect for Concurrent Copy.

Note:

1. Do not use an extent as the source in more than 12 FlashCopy relationships.
2. Do not use a FlashCopy target extent as a FlashCopy source extent in virtual-concurrent-copy.
3. An VCC, XRC, or PPRC source volume cannot become the FlashCopy target.
4. Working-space data sets can be allocated in the Extended Address Space (EAS) of an Extended Address Volume (EAV) if the level of z/OS supports the chosen dataset organization. That is, VSAM format working-space datasets can be allocated in the EAS on z/OS V1R10 or higher; non-extended sequential format working-space data sets can be allocated in the EAS on z/OS V1R12 or higher.

Virtual-concurrent-copy SMF type 42 subtype 4 records

The system data mover (SDM) writes a system management facility (SMF) type 42 subtype 4 record containing session statistics for each virtual-concurrent-copy session when the session ends. Virtual-concurrent-copy records contain the identifier VCC.

In the virtual-concurrent-copy SMF type 42 subtype 4 records, the following information are included:

- the size of the date that are used on CC and VCC. The following three fields shows the size:
  - Total tracks
  - CC tracks
  - VCC tracks
- the numbers of CC tracks that are read from disk and cache
- the maximum cache size for CC at a SSID boundary
- the total data space size for CC at a job name boundary
- the available WSDS space for VCC at a controller serial number boundary

ANTMIN00 parmlib

Concurrent copy and virtual concurrent copy support use of parmlib to control ANTMAIN. This support is modeled on the XRC support for ANTASnnn and ANTCLnnn address spaces documented in Chapter 6, “Administering your extended remote copy environment,” on page 123.

You specify the ANTMAIN control parameters that are applied to all systems IPLed using a given SYS1.PARMLIB concatenation in parmlib member ANTMIN00. SDM does not support selecting a specific member in the same way that SYSP can be used to select the IEASYSxx member.

You can specify ANTMAIN control parameters that apply to a specific system name in a component-specific parmlib dataset named *hlq* .ANTMAIN.PARMLIB, in a member name equal to the system name, for example SYS1.ANTMAIN.PARMLIB(SYSTEM1). In this way, systems can share a common SYS1.PARMLIB concatenation and still have different ANTMAIN configurations. For each parameter, the value found in *hlq* .ANTMAIN.PARMIB(*sysname*) takes precedence, followed by the value in the ANTMIN00 member of the SYS1.PARMLIB concatenation, followed by the default value if not specified.

If the parmlib data sets are protected by a security product such as RACF, ensure that the ANTMAIN address space has at least READ access authority to the data sets. Note that ANTMAIN needs this authorization even if your installation does not use concurrent copy or virtual-concurrent-copy.

Table 60. ANTMIN00 Parameters

Parameter	Value	Default
	Category: STARTUP	
CCAttnThrottle	String with one of these values: **YES** Throttles parallel attention behavior to a maximum of two parallel processes per controller session **NO** Does not restrict parallel attention processing per controller session. Aids in tuning the system to operate more efficiently during periods of high application updates.	YES
CCReadAhead	Integer in the range 3–64 Aids in tuning the system to operate more efficiently during periods of high application updates. This option allows you to set a maximum rate at which SDM reads updated track data from the controller cache. The integer value is the number of additional buffers obtained at the host for the read I/O for each concurrent copy session. Each buffer results in an additional 64K buffer being obtained from auxiliary storage and page fixed for the duration of the I/O request.	3
Hlq	String for the high level qualifier (up to 8 bytes)	SYS1
WSDSMsgMax	Integer in the range 1–10000 Maximum number of error messages to report to the console and syslog during working space dataset refresh processing, and as output from the LISTWS command. This value does not limit the number of lines reported by these functions when output is directed to a dataset.	10000

Table 60. ANTMIN00 Parameters (continued)

Parameter	Value	Default
AllowPPRConWSDS	String with one of these values: **NO** Working space data sets on PPRC primary volumes are marked unusable for VCC **YES** Working space data sets on PPRC primary volumes are eligible for use by VCC. If used, the behavior of the FlashCopy and PPRC operations depend on the installed storage control support. **PMPREF** If the WSDS is allocated on a Metro Mirror primary device, the FlashCopy Establish to the WSDS will indicate that the pair should not go into a duplex pending state as the result of this request. If a Preserve Mirror operation cannot be accomplished, the WSDS should still be used, allowing the pair to go into a duplex pending status as a result of the request **PMREQ** If the WSDS is allocated on a Metro Mirror primary device, the FlashCopy Establish to the WSDS will indicate that the pair must not go into a duplex pending state as the result of this request.	NO
	Category: FLAG	
NAME	String. The valid value is SMFVCC.	
ACTION	String. Use with NAME(SMFVCC). The values for ACTION are: **OFF** SDM suppresses the capture of VCC data used in the creation of SMF Type 42 Subtype 4 records. An SMF Type 42 Subtype 4 record may still be created when the session ends, but it will not contain data for the VCC or Snapshot operation. **ON** SDM captures data and produces SMF Type 42 Subtype 4 records for the VCC or Snapshot operation when the session ends, and they will contain VCC or Snapshot data.	OFF

Modify commands

SETPARMLIB - change the ANTMAIN settings

Purpose

Change the ANTMAIN address space settings from SYS1.PARMLIB(ANTMIN00). If you use the SETPARMLIB command to change the FLAG and PATCH parameters, the values of those flags and patches specified in the PARMLIB member are updated. Each time ANTMAIN's parameters are applied, the parmlib processor applies all of the flag and patch updates.

Command

SETPARMLIB

CCATTNTHROTTLE - control parallel attention processing

Purpose

Control parallel attention processing at the controller level:

YES
SDM should limit parallel attention processing to the number of parallel processes allowed at the controller level. This is the default.

NO SDM should not limit the number of parallel attention processes allowed at the controller level.

Use this to tune your system to operate more efficiently during periods of high application updates.

Command

F ANTMAIN,CCATTNTHROTTLE YES

CCREADAHEAD - tune the rate at which SDM reads updated track data from the controller cache

Purpose

Tune the rate at which SDM reads updated track data from the controller cache. The value is an integer in the range 3 through 64 that specifies the number of Read Track Image CCWs SDM builds over the actual count of CCWs needed to drain the controller cache sidefile. This value allows SDM to read additional data from the cache during peak update periods. The default is 3.

Use this to tune your system to operate more efficiently during periods of high application updates.

Command

F ANTMAIN,CCREADAHEAD *value*

CCSTOR - list the concurrent copy session status

Purpose

List concurrent copy sessions and the system resources the sessions are using.

Command

F ANTMAIN,CCSTORE

The following output shows that the concurrent copy jobs are running, but no changed tracks images are in SDM data space.

```
ANTX8969I NO CONCURRENT COPY JOBS
ANTX8969I NO CONCURRENT COPY TRACKS IN THE SIDEFILE DATA SPACE(S)
```

The following output shows that two concurrent copy jobs are running, and shows the total number of track images currently in SDM data spaces for each job.

```
ANTX8969I NAME     JOBID    ASID  % TRACKS
ANTX8969I -------- -------- ---- --- -------------
ANTX8969I DSSDUMPE JOB00020 002B 49 1,924
ANTX8969I DSSDUMPD JOB00019 0024 50 1,946
```

LISTWS - report the WSDS status

Purpose

Write a report of WSDS as system messages to the console or log, according to the following filter specification.

Table 61. F ANTMAIN,LISTWS filter Command

Filter	Result
INUSE	Shows WSDS that are in use by at least one job.
DRAINING	Shows WSDS for which a RELEASEWS is requested, or for an old HLQ.
COMPATIBLE	This filter takes an additional parameter, a list of VOLSERs to match WSDS to. The output is the list of WSDS that matches FlashCopy or FlashCopy compatibility with one or more of the input VOLSERs. The VOLSER list is a comma separated list of values enclosed in parenthesis. For example: `F ANTMAIN,LISTWS COMPATIBLE(PAS001,PAS002,PAS003)` The example shows all WSDS that are FC or FlashCopy compatible with at least one of the listed volumes.
ALL	Shows all WSDS. **Note:** Using the ALL filter might cause the message flood to the console or log.

Command

F ANTMAIN,LISTWS *filter*

The following screen is an example of the output received after entering the F ANTMAIN,LISTWS *filter* command.

```
ANTM6020I LISTWS filtertype OUTPUT – 001
ANTM6021I COMPATIBLE WITH VOLUMES (list of volsers, comma separated)
ANTM6022I WSDS PREFIX prefix
ANTM6023I NAME     STATUS JOBS RC   DS       CYL FREE CYL FLAGS
ANTM6024I --------------------------------------------------------
          nnnnnnnn ssssss jjjj rcrc dddddddd aaaaaaaa ffffffff
ANTM6025I FCWK0001 REL NA 6013 NOTAVAIL NOTAVAIL 04008120
```

PETINFO - display information about the SDM pause element token table

Purpose

Display information about the SDM Pause Element Token (PET) table, which is used when running DFSMSdss with Concurrent Copy, Fast Replication or extended-format data sets, as well when running applications that use the ANTRQST API.

Command

F ANTMAIN,PETINFO

The following sample output indicates that 5 of a maximum 20000 table entries are currently in use, and the high water mark for entries used was 507 at the date and time shown.

```
ANTM6997I USING 00005 OF 20000 PET. HWM 00507 AT 2008.302 06:41:05
```

Modifying the PET table size: Installations that use DFSMSdss to process a large amount of data with Concurrent Copy, Fast Replication, or extended format data sets may encounter errors indicating that the SDM PET table is full. If this occurs, installations may increase the size of the table from its default value of 20000 up to a maximum of 99999 using the following procedure:

1. Zap the ANTCTUNE CSECT of the ANTSDMLK load module in LINKLIB. The sample control statements shown below change the table size from 20000 to 40000.
```
NAME ANTSDMLK ANTCTUNE
VER 40 00004E20
REP 40 00009C40
```
2. Refresh LLA.
3. Issue the `F ANTMAIN,PETINFO` command to confirm that no table entries are currently in use.
4. Cancel the ANTMAIN address space. It will automatically restart, freeing the old table and allocating a new one with the new size. Note that any function that is currently using a PET table entry will encounter an error when ANTMAIN is cancelled. This includes both the DFSMSdss processing described above as well as any application that uses the ANTRQST API.

Each PET table entry requires 40 bytes of ECSA, so ensure that your system has sufficient available ECSA configured to accommodate the additional storage requirements.

REFRESHWS - refresh the WSDS list

Purpose

Refresh the WSDS list for VCC. By issuing the REFRESHWS command, you can see the number of WSDS, and the names of WSDS that are in use.

Command

F ANTMAIN,REFRESHWS

RELEASEWS - release ANTMAIN's enqueue on a WSDS

Purpose

Release ANTMAIN's enqueue on a WSDS. The command releases the enqueue on a WSDS by marking the WSDS as unavailable for new operations, and releasing the WSDS after the current operations complete. Parameter *llq* is the last qualifier of the dataset to be released, for non-VSAM data sets. For VSAM data sets, parameter *llq* is the last qualifier of the CLUSTER component of the data set. The DATA component is then selected for release automatically. For example, F ANTMAIN,RELEASEWS FCWK0004 releases work data set 4, either immediately if the data set is not in use, or at completion of all jobs using its space.

Command

F ANTMAIN,RELEASEWS *llq*

Part 8. Appendixes

Appendix A. Advanced Copy Services diagnostic aids

The following sections describe maintaining the system data mover (SDM) and diagnostic aids for PPRC and for the SDM functions XRC, SnapShot, and Concurrent Copy. Each SDM diagnostic operation indicates which components (XRC, CC, SnapShot, or the SDM itself) are applicable to that tool.

Maintaining the system data mover

Fixes and enhancements related to the system data mover are identified using SMP/E Enhanced HOLDDATA. Please refer to http://public.dhe.ibm.com/390holddata/390holddata.html for a description of using Enhanced HOLDDATA for z/OS.

Restarting system data mover address spaces (SDM, XRC, CC, SnapShot)

The XRC control address space, ANTAS000, and the concurrent copy address space, ANTMAIN, are automatically started during system IPL. The SDM automatically reinitializes either of these address spaces if the operator cancels it, or if either address space ends abnormally.

However, there are conditions that prevent the system from starting these address spaces at IPL time, or after an operator cancels the address spaces. For such cases, you can restart either (or both) of the ANTAS000 and ANTMAIN address spaces by submitting the following program:

```
//STARTANT JOB MSGLEVEL=(1,1),REGION=4096K
//STEP1    EXEC PGM=ANTSTRT
```

Diagnosing system data mover functions with the MVS MODIFY command

Use the MVS MODIFY command options that are described in this section to diagnose and repair XRC. The following command sections give descriptions for valid operations and operands:

Topic . . .
"RESTART operation (XRC)" on page 557
"SCDATA operation (XRC, CC)" on page 557
"SCTRAP operation (XRC)" on page 558
"STATESAVE operation (XRC)" on page 560
"SUSSESS operation (XRC)" on page 561
"TERMDVC operation (XRC)" on page 562
"TERMSESS operation (XRC, CC)" on page 562
"XENDDUMP operation (XRC)" on page 564
"XRCTRAP/SUPRDUMP operation (XRC)" on page 564

Note: The publication *z/OS MVS System Messages, Vol 1 (ABA-AOM)* describes the system data mover messages (which begin with ANT) and error codes. Diagnostic procedures outlined in this section generate messages that begin with ANTX89.

If the messages are insufficient to diagnose your problem, issue the XQUERY TSO command to determine the status of the copy operation and volume pairs. If you need to report your problem to IBM for further diagnosis, please include the following information with your report:

- Merged console log from all attached systems. The SYSLOG contains a record of all XRC commands and messages, except for TSO commands.
- Merged SYS1.LOGREC data sets from all attached systems.
- For I/O related errors, a GTF trace of the failing command sequence on both the primary and secondary volumes.
- For functional errors, generate an SVC dump of the SDM address spaces ANTMAIN, ANTAS000, and ANTAS*nnn*. The value for the address space indicated by the ANTAS*nnn* variable can be ANTAS000 or any of the address spaces numbered ANTAS001–ANTAS020.

  Note: To determine which ANTAS*nnn* address space is associated with the session, look at message ANTQ8200I that appears in the header of XQUERY reports for the session.

Attention: The MODIFY command is abbreviated as F. The format of the MODIFY command is as follows:

```
F ANTASnnn,operation (optional_operands)
```

CREFRESH operation (XRC)

Use the CREFRESH operation to refresh a cluster session and its associated address space. To enable a cluster address space after setting up the XRC PARMLIB parameter of ClusterMSession and optionally ClusterName, issue the following MVS MODIFY command:

```
F ANTAS000,CREFRESH
```

You may use this operation to clear up a cluster session. Specify the optional parameter of FORCE with this operation:

```
F ANTAS000,CREFRESH FORCE
```

Note: FORCE does not remove the cluster address space. It cleans up the XRC control blocks and enables the cluster address space to be started again by using the normal CREFRESH command. To remove the cluster address space, use the MVS operator CANCEL command.

You may use this operation to disable clustering. Specify the optional parameter of DISABLE with this operation:

```
F ANTAS000,CREFRESH DISABLE
```

DISABLE causes the same action as if you had coded ClusterMSession(disabled) in the parmlib, then issued a normal CREFRESH. The value of ClusterMSession in the parmlib is not changed.

To use the CREFRESH command to start a cluster other than the one specified on the ClusterName parameter in the XRC PARMLIB, issue the following command:

F ANTAS000,CREFRESH *clustnam*

Note that if a cluster name is specified on the CRFERESH command, all other cluster-related attributes (such as ClusterMSession) remain the same.

DELBMAP operation (XRC)

This operation requests SDM to delete all of the bit maps for ESS controllers in the state data set. The state data set belongs to the address space the command executes in. The command does not run in the control address space (ANTAS000).

Example: Issuing the following command deletes all ESS bit maps in the state data set for primary address space 002:

```
F ANTAS002,DELBMAP
```

DUMP operation (XRC, CC)

Used primarily for debugging purposes, this operation obtains a dump of the XRC system data mover address space and the associated trace data space. The SYS1.DUMP data set receives the output.

Examples: Examples of the dump operation follow:

- In order to dump the address space and trace data space for the XRC system data mover function, use the command:

  ```
  F ANTAS003,DUMP
  ```

- In order to dump the address space and trace data space for concurrent copy, use the command:

  ```
  F ANTMAIN,DUMP
  ```

The DUMP operation does not affect the current state of the XRC SDM function. Dumping the SDM does, however, delay record update processing, and therefore increases the time until the dump operation is complete.

DVCDATA operation (XRC, CC)

This operation reports the storage control sessions that exist for a device. You can perform this operation regardless of whether XRC or concurrent copy is active on the system. You may only issue the MODIFY command with the DVCDATA operation from a system which has the ANTAS000 address space active.

Specify the device address for the device for which you want session information. The storage control Licensed Internal Code (LIC) must at least support concurrent copy.

Example: Issue the following command to list the existing sessions for the device F40:

```
F ANTAS000,DVCDATA F40
```

Result: Possible output in response to this command would be:

```
ANTX8965I F40 XA01
```

This means that storage control session 1 is an active XRC session.

The operation output contains: the system data mover message number, ANTX8965I; the specified device (in this case, it is F40); and an individual session report about each session for the specified device (in the example, it is XA01), which includes the following information:

- Type of storage control session
- Status of storage control session
- Storage control session number
- Host connection status

This information for each session has a format of *tsnnc*, where the following apply:

t
: C = concurrent copy session, X = XRC session.

s
: A = active session, S = suspended session, Q = quiesced session, T = timed-out session.

nn
: The storage control session number.

c
: The connection status of the storage control session, as known by the system that issued the command. If *c* is blank, this host processor owns the session. If *c* is “*”, another processor owns the session. If *c* is “?”, this host processor owns the session, but may or may not have access to the session.

Results: Typical responses for this command are:

- ANTX8965I F40 XA01—This means that storage control session 1 is an active XRC session.
- ANTX8965I F40 CA02—This means that storage control session 2 is an active concurrent copy session.

- ANTX8965I F40 - CA03 XQ04—This means that there are two storage control sessions: Session 3 is an active concurrent copy session, and Session 4 is a quiesced XRC session.
- ANTX8966I F40 NO SYSTEM DATA MOVER STORAGE CONTROL SESSIONS—This means that there are no sessions for the storage control that is associated with device F40.

Note: If errors occur, this command might also issue I/O error messages.

IGNORE_INTERRUPTS operation (XRC)

This operation requests that the data mover ignore low attention threshold and high attention threshold interrupts that are sent by a storage control. If **ON** is specified, the data mover ignores the low and high attention interrupts. The data mover uses the ReadDelay value that is specified in the PARMLIB to determine how often to read record sets from the storage control. This value might need to be adjusted if the low and high attention interrupts are being ignored. The ON option might have the affect of reducing the amount of CPU time that is used by the data mover. If the IGNORE_INTERRUPTS command is specified without an operation, the current setting is returned.

Example: Issue the following command to activate the IGNORE_INTERRUPTS function:

```
F ANTAS001, IGNORE_INTERRUPTS ON
```

Issue this command using the OFF option to disable the function.

LISTDVCS operation (XRC)

This operation lists all of the devices that are part of a specific storage control session. A storage control session is comprised of all of the devices, on a particular storage control, that contain volumes that belong to a unique XRC session. The first device listed is the one that was used to initially set up the session.

The command syntax is as follows:

```
F ANTAS000,LISTDVCS dddd ss
```

The following definitions apply to the example above:

dddd The address of a device that is attached to the primary storage control you want to display. The storage control must have XRC-capable Licensed Internal Code (LIC) that supports this function.

ss The storage control session number (which you can obtain by issuing a F ANTAS000,LISTSESS *xxxx* command—see “LISTSESS Operation” on page 554 for more information).

Example: Issue the following command to list the devices owned by storage control session 04 of the storage control for device F40:

```
F ANTAS000,LISTDVCS 0F40 04
```

Result: The following is a possible response to this command:

```
ANTX8927I 0F40 – XA04 – 25(0F40) 0F(....)
```

This means that session 04 is an active storage control session that has two devices. Device 0F40 relates to device channel connection address 25. The device that relates to device channel connection address 0F is offline.

The operation output contains: the system data mover message number, ANTX8927I; the address of the device (which is 0F40); information about the session (see "LISTSESS Operation" for more information about the session status syntax); and multiple fields that provide information about each session for the specified device. The information for each session has a format of *cc(dddd)s*, where the following apply:

cc
The channel connection address (CCA).

dddd
The device number. If the device is offline, the value is "....".

s
An "s" indicates that the session is suspended. A blank indicates that the session is not suspended. This field is reported for ESS storage subsystems, only.

Results: The following are possible responses to the command:

- `ANTX8927I 0F40 – XA04 – 25(0F40) 0F(....)`—This means that session 04 is an active storage control session that has two devices. Device 0F40 relates to device channel connection address 25. The device that relates to device channel connection address 0F is offline.
- `ANTX8928I 0F40 NO DEVICES FOR STORAGE CONTROL SESSION 04`—This means that storage control session 04 is not in the storage control.

LISTSESS Operation

This operation reports the storage control sessions that exist for a storage control. You can perform this operation regardless of whether XRC or concurrent copy is active on the system. You can only issue the MODIFY command with the LISTSESS operation from a system which has the ANTAS000 address space active.

Specify a device address for a device that is attached to the storage control for which you want session information. You can also specify the word ALL to obtain information about all the sessions in the system. The storage control LIC must at least support concurrent copy.

LISTSESS, LSTDVCS, DVCDATA, and SCDATA may now all display 'E' for the storage control session type, representing an XRC enhanced (multiple reader) session.

When using LISTSESS with a device address, you may also include the E parameter, to obtain a display of the XRC enhanced (multiple reader) session structure.

Example 1:

```
F ANTAS000,LISTSESS 0F76 E
```

Could produce this response:

```
ANTX8913I 0F76 - 01 - 03 -04
```

where 01 is the base storage control session, and 03 04 are auxiliary storage control sessions, in the enhanced (multiple reader) session group.

If there is no enhanced session group associated with the storage control for the specified device, the following message will be issued:

```
ANTX8914I 0F76 NO SYSTEM DATA MOVER STORAGE CONTROL SESSIONS
```

Note: There may still ne non-enhanced sessions associated with the storage control. This can be displayed by reissuing the LISTSESS command without the E parameter.

When using TERMSESS to terminate the base storage control session in an enhanced (multiple reader) session group, you must first terminate all auxiliary storage control sessions. An attempt to terminate a base session that has an associated auxiliary session will fail with the following message:

```
ANTX8987I OUTSTANDING AUXILIARY SESSIONS
```

Example 2:

As an example, issue the following to list the existing sessions for the storage control that is attached to device F40:

```
F ANTAS004,LISTSESS F40
```

Result: The following is a possible response to this command:

```
ANTX8913I F40 XA01
```

The response means that storage control session 1 is an active XRC session.

You can also issue the following command to list all the sessions in the system:

```
F ANTAS000,LISTSESS ALL
```

Result: All the sessions that exist in the system are shown in ANTX8193l messages. Refer to *z/OS MVS System Messages, Vol 1 (ABA-AOM)* for details. Sessions for both online and offline devices are displayed. Offline devices are displayed even if they have never been online and/or are associated with sessions on another system.

The operation output contains multiple fields that provide information about each session on the specified device's storage control. The command returns the following information for each session:

- Type of storage control session
- Status of storage control session
- Storage control session number
- Host connection status

An individual session report has a format of *tsnnc*, where the following apply:

t C = concurrent copy session, X = XRC session.

s A = active session, S = suspended session, Q = quiesced session, T = timed-out session.

nn The storage control session number.

c The connection status of the storage control session, as known by the system that issued the command. If *c* is blank, this host processor owns the session. If *c* is "*", another processor owns the session. If *c* is "?", this host processor owns the session, but may or may not have access to the session.

Results: Other possible responses to this command are:

- `ANTX8913I F40 CA02`—This means that storage control session 2 is an active concurrent copy session.
- `ANTX8913I F40 - CA03 XQ04`—This means that there are two storage control sessions: Session 3 is an active concurrent copy session, and Session 4 is a quiesced XRC session.
- `ANTX8914I F40 NO SYSTEM DATA MOVER STORAGE CONTROL SESSIONS`—This means that there are no sessions for the storage control that is associated with device F40.
- `ANTX8914I ALL NO SYSTEM DATA MOVER STORAGE CONTROL SESSIONS`— This means that there are no sessions in the system.
- `ANTX8980I FINISHED LISTSESS ALL PROCESSING` — This means the LISTSESS ALL command has finished processing.

Note: If errors occur, this command might also issue I/O error messages.

PATHS operation

This operation reports the channel path and associated path group IDs for a device. You can perform this operation regardless of whether XRC or concurrent copy is active on the system. You can issue the MODIFY PATHS command from a system which has the ANTAS000 address space active or from any ANTAS*nnn* active address space.

Example: Issue the following command to activate the PATHS function:

```
F ANTAS000, PATHS ddd
```

ddd indicates a device number.

Result: The following is an example output when you specify the F40 device number:

```
F ANTAS001,PATHS F40

ANTX8977I 0F40 - CPID=**  PGID=000006064C2064B8F76671
ANTX8977I 0F40 - CPID=52  PGID=0000000000000000000000
ANTX8977I 0F40 - CPID=53  PGID=0000000000000000000000
ANTX8977I 0F40 - CPID=7E  PGID=000006064C2064B8F76671
ANTX8977I 0F40 - CPID=7F  PGID=000006064C2064B8F76671
```

This output has the following meaning:

- 0F40 is the device number
- CPID is the channel path ID

- PGID is the path group ID
- The line with the CPID of '**' shows the initial PGID that z/OS sends to the storage control
- A PGID of all zeros indicates one of the following conditions:
  - The path group is physically unavailable
  - All the devices associated with the channel path were offline when the channel path was varied online

REDISCOVER operation (XRC)

This operation forces a rediscovery of offline devices on the XRC LPAR. This may be useful following dynamic I/O reconfiguration when rediscovery can shorten the time that subsequent XRC commands take to process. You can issue the MODIFY command with the REDISCOVER operation only from a system that has the ANTAS000 address space active.

Example: The following example shows the REDISCOVER operation:

```
F ANTAS000,REDISCOVER
```

RESTART operation (XRC)

The format of the RESTART command is as follows:

```
F ANTAS005,RESTART
```

This is a serviceability function that forces the XRC session to restart with minimal affect to the storage control session. Use this operation when it appears that the system data mover has stalled or is not responding in the expected manner. The RESTART command, which you can only direct to the ANTAS*nnn* address space associated with the session you want to restart, performs the following actions:

1. Generates an SVC dump of the ANTAS*nnn* address space.
2. Saves the software bitmaps into the state data set for all volumes in the XRC session that are attached to storage controls that support hardware bitmapping.
3. Suspends the XRC session and ends the ANTAS*nnn* address space.
4. Restarts the XRC session. When the ANTAS*nnn* address space has restarted, the SDM begins to drain updated records from the cache for storage controls that do not support hardware bitmaps.

After the XRC session has restarted, you can issue XADDPAIR commands to bring all of the suspended volumes back into the session. Storage controls that support hardware bitmapping resynchronize the volumes from the stored bitmap information.

SCDATA operation (XRC, CC)

This operation reports the status of a session on a primary storage control. You may perform this operation regardless of whether XRC or concurrent copy is active on the system. You can only issue the MODIFY SCDATA command from a system which has the ANTAS000 address space active.

The following example is the command syntax:

```
F ANTAS002,SCDATA ddd ss X
```

The following definitions apply to the syntax above:

ddd
: Specifies the address of a device that is attached to the primary storage control you want to display. The storage control must have XRC-capable Licensed Internal Code (LIC) that supports this function.

ss
: Specifies the storage control session number (which you can obtain by issuing a F ANTAS000,LISTSESS *xxxx* command, see "LISTSESS Operation" on page 554 for more information).

X
: Specifies that the channel extender status for a device be provided by way of the ANTX8970I message.

Result: If you specify the X parameter, the following example message displays the channel extender status for the device:

```
ANTX8970I F40 CHANNEL EXTENDER TYPE=CNT LEVEL=01 ALLOC=4000
    BUFFER=000B0000
```

Result: If there is no channel extender, the following message appears:

```
ANTX8970I 0F40 NO CHANNEL EXTENDER
```

Without the X parameter, the command returns a system data mover message of number ANTX8936I.

Example: If you issue the following command to list the current status of session 01 on the primary storage control that is associated with device F40:

```
F ANTAS002,SCDATA 0F40 01
```

Result: An example of the output is:

```
ANTX8936I 0F40 - SESSION=XA01 RESIDUAL=0005 NEXTDEV=0F40 MAXSIZE=2048
                 TIME(1997.234 12:24:32.273568) TIMEOUT(00.05.00)
```

The following is a sample message produced by the SCDATA command:

```
ANTX8936I 0F60 - SESSION=XA09 RESIDUAL=0000 NEXT DEV=0F60* MAXSIZE=E000
TIME(************************) TIMEOUT(00.05.00)
```

Note: For a description of the ANTX8936I message, refer to *z/OS MVS System Messages, Vol 1 (ABA-AOM)*.

SCTRAP operation (XRC)

This operation is a serviceability function provided by the XRC support. The SCTRAP operation enables a storage control to dump its internal control queues when requested to do so by an XRC function. This information is useful for diagnosing the storage control errors for the storage controls that support this capability.

This operation applies to the primary storage controls associated with an XRC logical session. XRC only requests one storage control dump each 24-hour period.

To obtain additional dumps within this time period, use the MVS MODIFY STATESAVE RESET option to change the last time when a dump was taken to zero.

Note: This operation does not apply to non-disruptive state saves.

Example: The following example is the SCTRAP operation:

```
F ANTAS003,SCTRAP ON
```

Specifying the ON operand enables XRC to request a storage control to dump its internal control queues when it encounters certain LIC-related errors. Specifying the OFF operand tells XRC, for certain error types, not to request a storage control to dump its internal control queues.

If you do not specify an operand, MVS displays the current status of the SCTRAP operation.

Note: Turning SCTRAP on does not cause the storage control to dump its queues. In the following situations, turning SCTRAP on only enables XRC to make the request:
- The storage control encounters one of the predefined error conditions.
- A STATESAVE operation with device address and session number option is executed.

There can be an application performance impact when the system dumps the storage control queues.

SCTRAP2 operation (XRC)

This operation is a serviceability function provided by XRC support. The SCTRAP2 option enables a storage control to dump its internal control queues when requested to do so by an XRC function. This information is useful for diagnosing the storage control errors for the storage controls that support this capability.

This operation applies to the secondary storage controls associated with an XRC logical session. XRC only requests one storage control dump each 24-hour period. To obtain additional dumps within this time period, use the MVS MODIFY STATESAVE RESET2 option to change the last time when a dump was taken to zero.

Note: This operation does not apply to non-disruptive state saves.

Example: The following example is the SCTRAP2 operation:

```
F ANTAS003,SCTRAP2 ON
```

Specifying the ON operand enables XRC to request a storage control to dump its internal control queues when it encounters certain LIC-related errors. Specifying the OFF operand tells XRC, for certain error types, not to request a storage control to dump its internal control queues.

If you do not specify an operand, MVS displays the current status of the SCTRAP2 operation.

Note: Turning SCTRAP2 on does not cause the storage control to dump its queues. Turning SCTRAP2 on only enables XRC to make the request if it encounters one of the predefined error conditions. There can be an application performance impact when the system dumps the storage control queues.

STATESAVE operation (XRC)

This operation has several options provided by XRC support. The options are: device address and session number, SHOW, SHOW2, RESET, and RESET2.

device address/*session number*/**N**

This option directs a storage control to dump its internal control queues, called a state save.

device address

Tells XRC which storage control to direct the request to. Devices defined in an alternate subchannel set can be specified by coding a 5-digit device address, where the leading digit indicates the subchannel set in which the device is physically defined. If you specify fewer than 5 digits, the command is issued to the device that is currently logically in subchannel set 0. If a HyperSwap has occurred, this could be the device that was physically defined in an alternate subchannel set.

session number

Is the XRC storage control session number. If a number is not applicable, use zero.

N Requests a non-disruptive state save, that is, a state save without warmstart. If the storage control microcode doesn't support a non-disruptive state save, and the setting for SCTRAP is ON, a standard state save is taken. If a non-disruptive state save is supported, the setting for SCTRAP is ignored.

Example: The following example is the *device address/session number* operation:

```
F  ANTAS000,STATESAVE  F40  4
```

SHOW

The SHOW option directs XRC to show the timestamp of the last state save. The state save is taken for the primary storage controls associated with a logical XRC session.

Example: The following example is the SHOW operation:

```
F ANTAS003,STATESAVE SHOW
```

SHOW2

The SHOW2 option directs XRC to show the timestamp of the last state save. The state save is taken for the secondary storage controls associated with a logical XRC session.

Example: The following example is the SHOW2 operation:

```
F ANTAS003,STATESAVE SHOW2
```

RESET

The RESET option directs XRC to reset to zero for the timestamp of the last state save. The state save is taken for the primary storage controls associated with a logical XRC session.

Example: The following example is the RESET operation:

```
F ANTAS003,STATESAVE RESET
```

RESET2

The RESET2 option directs XRC to reset to zero for the timestamp of the last storage control state save. The state save is taken for secondary storage controls associated with a logical XRC session.

Example: The following example is the RESET2 operation:

```
F ANTAS003,STATESAVE RESET2
```

SUSSESS operation (XRC)

The SUSSESS operation provides the ability to suspend an XRC scsession from any application host system. This provides an installation with the control to suspend a session even if the data mover system is unable to communicate to the primary storage controls. The SUSSESS operation allows you to place an XRC session in hardware bitmap mode. This means that when the XRC session resumes, only the changed tracks must be re-copied. This provides a significant reduction in resynchronization time, which helps maintain the Recover Point Objective time.

Use the SUSSESS operation to suspend an XRC scsession from any system that has access to the device.

Example: Issue the following command to initiate the SUSSESS function:

```
F ANTAS000,SUSSESS DEVICE_NUMBER SCSESSION_ID
```

The following definitions apply to the example above:

device_number

Specifies the device that is part of the storage control session that you want to control.

scsession_id

Specifies the ID of the storage control session that you want to suspend.

You can also perform the SUSSESS operation using the ANTRQST assembler macro. Use the DEVN and SCSESSION parameters with the XSUSPEND command to perform this request. The DEVN parameter provides the device number of the storage control session that you want to suspend. The SCSESSION parameter provides the scsession ID of the storage control session that you want to suspend. Complete documentation for these parameters is found at the front of the macro description for ANTRQTA in SYS1.MACLIB.

After the SUSSESS operation has been initiated, XRC returns either a return code 630 or return code 647 error. The error code depends on the timing of when the next read record set channel program is initiated.

If desired, the device_number and scsession_id may be replaced with the word **ALL**. This will cause all active XRC sessions in the system to be suspended. At least one device for each scsession must be online to the MVS system where the command is issued.

Example: Issue the following command to initiate the SUSSESS function using the ALL parameter:

```
F ANTAS000,SUSSESS ALL N
```

where *N* indicates that verification of the request is not to be issued.

TERMDVC operation (XRC)

This operation ends the relationship between a device and an active XRC session for a storage control.

Guidelines: Follow these guidelines to use the TERMDVC operation:

- You can only issue the operation from a system that has the XRC function installed and has the ANTAS000 address space active.
- You cannot issue this operation if the device is the last device that is associated with the XRC session—the operation will not complete.
- You must issue the operation from the processor where the XRC storage control session is active, or was last active. If you cannot issue the command from that processor, use the TERMSESS operation to end the XRC session (see "TERMSESS operation (XRC, CC)").

You must specify the following information with the TERMDVC operation:

- A device address. The LIC in the device's storage control must at least support XRC and the TERMDVC function. You can use LISTDVCS to show the devices that are associated with the XRC session. The device address must be associated with the XRC session (see "LISTDVCS operation (XRC)" on page 553 for more information).
- The session number that is associated with the volume whose relationship you want to end.

Example: Issue the following command to end the relationship between device F40 and storage control session 4:

```
F ANTAS004,TERMDVC F40 4
```

TERMSESS operation (XRC, CC)

This operation ends an active, quiesced, or timed-out XRC or concurrent copy session for a storage control. You may only issue the operation from a system that has the XRC function installed, and has the ANTAS000 address space active. TERMSESS provides several kinds of support, shown in Table 62 on page 563.

Table 62. Using the TERMSESS operation

If you want . . .	Then . . .	Notes . . .
To end a single session	Specify: • A device address. For a session that is in either active status or timed-out status, you must issue the operation from the processor where the XRC session is active, or was last active. For a quiesced or suspended session, issue the operation from any processor. • The storage control session number of the session you want to end. **Example:** To end active storage control session number four, using device F40, use: `F ANTAS005,TERMSESS F40 4`	The LIC in the device's storage control must at least support concurrent copy. The device address must belong to the session that you plan to end, and the device must be online to the processor that you are issuing the TERMSESS operation from. You can use LISTDVCS to show the devices that belong to the session.
To end all active or timed-out XRC sessions that were last active on the processor from which you issued the TERMSESS operation	Specify XRC. The TERMSESS XRC operation issues a verify prompt (REPLY Y | N) for the operation. You will also receive a verify prompt (REPLY Y | N) for each eligible storage control session unless you specify N with the operation. **Example:** For example, to end all active or timed-out XRC storage control sessions, and *not* be prompted for each individual session, use: `F ANTAS005,TERMSESS XRC N`	At least one device that is associated with the XRC session must be online to the processor from which you issue the TERMSESS operation.
To end all quiesced or suspended XRC sessions, regardless of which processor the XRC session was last active on	Specify SUS. The TERMSESS SUS operation issues a verify prompt (REPLY Y | N) for the operation. You will also receive a verify prompt (REPLY Y | N) for each eligible storage control session, unless you specify N with the operation. **Example:** For example, to end all quiesced or suspended XRC storage control sessions with prompting for each individual session, use: `F ANTAS005,TERMSESS SUS`	At least one device that is associated with the XRC session must be online to the processor from which you issue the TERMSESS operation.
To end all XRC or concurrent copy sessions that are in timed-out status	Specify TIM. The TERMSESS TIM operation issues a verify prompt (REPLY Y | N) for the operation. You will also receive a verify prompt (REPLY Y | N) for each eligible storage control session, unless you specify N with the operation. **Example:** To end all timed-out XRC and concurrent copy storage control sessions, with prompting for each individual session, use: `F ANTAS005,TERMSESS TIM`	At least one device that is associated with the XRC session must be online to the processor from which the TERMSESS operation is issued.

Table 62. Using the TERMSESS operation (continued)

If you want . . .	Then . . .	Notes . . .
To end all Concurrent Copy sessions on all storage controls that are visible to the system from which the command is issued	Specify CCS. Add the N option to request no confirmation prompts. **Example:** To end global Concurrent Copy sessions without prompts, use: `F ANTAS000,TERMSESS CCS N`	The sessions being ended may include: • Sessions for devices that are not currently online to the issuing system • Sessions for currently running jobs. You should expect those jobs to fail as a result of the session termination.
To have the command be effective from any device in the same storage control as the target session	Specify FORCE. **Example:** To end storage session 02 on the storage control where device E023 resides, use: `F ANTAS000,TERMSESS E023 02 FORCE`	This also allows the command to be issued from any system that has access to the storage control, and not just from the system that created the session. FORCE requires storage controls with Licensed Internal Code that supports the function.

XENDDUMP operation (XRC)

This is a serviceability function that forces XRC to generate a dump whenever a session-level XSUSPEND or XEND command is issued. XENDDUMP obtains diagnostic information before the ANTAS*nnn* address space ends. You might typically issue the command to capture diagnostic information during problem re-creation. Issue the MODIFY command with the XENDDUMP operation only from the ANTAS*nnn* address space. If you issue it from the ANTAS000 address space, XRC does not generate a dump.

Specifying the ON operand directs XRC to generate a dump whenever an XEND or XSUSPEND(session) command completes. Specifying the OFF operand directs XRC to not generate a dump. If you do not specify an operand, MVS displays the current status of the XENDDUMP operation.

Example: An example of this command follows:

```
F ANTAS001,XENDDUMP ON
```

XRCTRAP/SUPRDUMP operation (XRC)

The commands XRCTRAP and SUPRDUMP both perform the serviceability function that controls whether XRC generates a dump as a result of a software error.

Specifying the OFF operand on either command directs the system data mover, if required, to dump to the SYS1.DUMP data set when it encounters a software error. Specifying the ON operand on either command suppresses dumps on software-related errors. The default is OFF.

If you do not specify an operand, MVS displays the current status of the XRCTRAP or SUPRDUMP operation.

Example: An example of the XRCTRAP operation follows:

```
F ANTAS002,XRCTRAP ON
```

Example: An example of the SUPRDUMP operation follows:

```
F ANTAS004,SUPRDUMP OFF
```

Peer-to-peer remote copy diagnostic aids

Peer-to-peer remote copy (PPRC) is a hardware feature of cached storage controls. TSO software operations enable you to control PPRC operations. ERP messages indicate when PPRC has suspended a volume pair, as well as the time and the reported hardware indicator for the suspension. The publication *z/OS MVS System Messages, Vol 1 (ABA-AOM)* describes the PPRC TSO and ERP messages.

If the messages are insufficient to diagnose your problem, issue the CQUERY TSO command to determine the status of the copy path and volume pair. If you need to report your problem to IBM for further diagnosis, please include the following information with your report:

- Merged console log from all attached systems. The SYSLOG contains a record of all PPRC operations and messages.
- Merged SYS1.LOGREC data sets from all attached systems.
- For I/O related errors, a GTF trace of the failing operation sequence for pairs involved with the error.

Appendix B. SMF type 42 records

The SMF type 42 records for extended remote copy (XRC) and concurrent copy (CC) sessions are summarized below. *z/OS MVS System Management Facilities (SMF)* contains the complete layout of SMF type 42 records.

XRC information in SMF type 42 records

The system data mover (SDM) writes a System Management Facilities (SMF) record type 42 subtype 11 for extended remote copy session statistics whenever the SMF timer interval ends. This SMF XRC information can be used by the DFSMSdfp Optimizer to provide information for capacity planning and environmental tuning.

Note: XRC cannot process SMF records for physical sessions that have ended during the SMF interval. A session can be ended, for instance, due to a “lost” physical session.

The following three fields are included in the SMF type 42 record when a subtype 11 record is written:

Offsets Dec.	Hex	Name	Length	Format	Description
36	24	SMF42XRO	4	binary	Offset to XRC statistics.
40	28	SMF42XRL	2	binary	Length of XRC statistics.
42	2A	SMF42XRN	2	binary	Number of XRC sessions.

Table 63 contains the SMF record type 42 information that applies to XRC. This information is grouped within a subtype 11 record, which is only included when there is an active XRC session. A volume must be in duplex state before SMF information is accumulated for that volume.

Table 63. Record layouts for subtype 11

Offsets Dec.	Hex	Name	Length	Format	Description
The following relate to an XRC session:					
0	0	S42XRID	8	EBCDIC	Logical session ID.
8	8	S42XRTYP	8	EBCDIC	Session type: *XRC* = disaster recovery session, *MIGRATE* = migration session.
16	10	S42XRSSO	4	binary	Offset to first storage subsystem identifier (SSID) data section.
20	14	S42XRSSN	2	binary	Number of SSIDs for the session.
22	16	S42XRSSL	2	binary	Length of SSID data section.
24	18	S42XRVLO	4	binary	Offset to first volume data section.
28	1C	S42XRVLN	2	binary	Number of volume data sections.
30	1E	S42XRVLL	2	binary	Length of volume data sections.
The following relate to XRC SSID data:					
0	0	S42XRSNX	4	binary	Offset to next SSID data section (0 if last SSID).

Table 63. Record layouts for subtype 11 (continued)

Offsets Dec.	Hex	Name	Length	Format	Description
4	4	S42XRSID	2	binary	SSID.
6	6	S42XRIDP	1	binary	Storage control session ID.
7	7		1		Reserved.
8	8	S42XRVSH	4	binary	Number of primary volumes associated with this SSID that are still included in the XRC session at the end of the SMF interval. This includes both volume pairs and utility volumes.
12	C	S42XRTPR	4	binary	Total number of primary SDM record updates read.
16	10	S42XRNWD	4	binary	Number of primary SDM reads with data.
20	14	S42XRNND	4	binary	Number of primary SDM reads with no data.
24	18	S42XRNLR	4	binary	Number of primary SDM record updates left to be read at the point in the SMF when information is collected.
28	1C	S42XRNFW	4	binary	Number of format writes.
32	20	S42XRNUW	4	binary	Number of update writes.
36	24	S42XRARS	4	binary	Average record update size of data processed, in bytes.
40	28		4		Reserved.
The following relate to XRC volumes:					
0	0	S42XRVLX	4	binary	Offset to next volume data section (0 if last volume).
4	4	S42XRVLV	6	EBCDIC	Volume serial number.
10	A	S42XRVAV	6	EBCDIC	Other volume serial number in pair.
16	10	S42XRVPS	1	EBCDIC	“P” if primary volume, “S” if secondary volume, “U” if utility volume.
17	11		1		Reserved.
18	12	S42XRVSS	2	binary	SSID associated with volume.
20	14	S42XRVPR	4	binary	Primary = total number of primary SDM record updates read. Secondary = total number of SDM records written.
24	18	S42XRVWD	4	binary	Primary = number of SDM reads with data. Secondary = 0.
28	1C	S42XRVND	4	binary	Primary = number of SDM reads with no data. Secondary = 0.
32	20	S42XRVFW	4	binary	Number of format writes.
36	24	S42XRVUW	4	binary	Number of update writes.
40	28	S42XRVRS	4	binary	Primary = average record size read, in bytes. Secondary = average record size written, in bytes.
44	2C	S42XRVRW	4	binary	Primary = number of read tracks. Secondary = number of write tracks.
48	30		4		Reserved.

Concurrent copy information in SMF type 42 records

The system data mover (SDM) writes a system management facility (SMF) type 42 subtype 4 record containing session statistics for each concurrent copy session when the session ends. Concurrent copy records contain the identifier *CC*.

The subtype 4 header section begins at offset 36 in the SMF record and contains the following fields. (The fields contain zeros if the SMF record was written for an extended sequential data set.)

Offsets Dec.	Hex	Name	Length	Format	Description
36	24	SMF42CCO	4	binary	Offset to CC statistics.
40	28	SMF42CCL	2	binary	Length of CC statistics.
42	2A	SMF42CCN	2	binary	Number of CC session.

Concurrent copy SMF type 42 subtype 4 records

The concurrent copy header section starting at the offset contained in SMF42CCO provides information about the concurrent copy session:

Table 64. Record layouts for subtype 4

Offsets Dec.	Hex	Name	Length	Format	Description
The following relate to a concurrent copy (CC) session:					
0	0	S42CCID	4	binary	Logical session ID.
4	4	S42CCRQS	2	EBCDIC	Session type: *CC* indicates concurrent copy.
6	6	S42CCTS	1	EBCDIC	Termination status of the session (*N* = normal termination, *A* = abnormal termination).
7	7		1		Reserved.
8	8	S42CCJNM	8	EBCDIC	Jobname that requested the CC session.
16	10	42CCJNO	8	EBCDIC	JES number of the job that requested the CC session.
24	18	S42CCSST	8	binary	The time (TOD timestamp) that the CC session started.
32	20	S42CCEIT	8	binary	The TOD that initialization completed for the CC session.
40	28	S42CCSET	8	binary	The TOD that the concurrent copy session ended.
48	30	S42CCSSO	4	binary	Offset to header information for the first subsystem in the CC session.
52	34	Schussing	2	binary	Number of subsystems involved in the CC session.
54	36	S42CCSSL	2	binary	Length of the SSID header.
56	38		8		Reserved.
The following relate to the storage subsystem:					
0	0	S42CSNXT	4	binary	Offset to the next SSID header (0 if last SSID).
4	4	S42CSID	2	binary	SSID.
6	6	S42CSIDP	1	binary	Storage control session ID.
7	7		1		Reserved.

Table 64. Record layouts for subtype 4 (continued)

Offsets Dec.	Hex	Name	Length	Format	Description
8	8	S42CSMSF	4	binary	Maximum track threshold that was held in the storage control sidefile for the CC session.
12	C	S42CSVLO	4	binary	Offset to first volume in the CC session for this SSID.
16	10	S42CSVLN	2	binary	Number of volumes in the CC session associated with this SSID.
18	12	S42CSVLL	2	binary	Length of the volume section.
20	14		4		Reserved.
The following relate to volumes associated with the SSID in the previous section:					
0	0	S42CVLNX	4	binary	Offset to next volume for the subsystem (0 if last volume for the subsystem).
4	4	S42CVLSR	6	EBCDIC	Volume serial number for the volume.
10	A		2		Reserved.
12	C	S42CVLDV	1	binary	Device type (UCBTBYT4) for the volume.
13	D	S42CVLUA	3	EBCDIC	Unit address (UCBNAME) for the volume.
16	10	S42CVLTK	4	binary	Number of tracks DFSMSdss requested to be processed on the volume.
20	14	S42CVLRD	4	binary	Number of tracks read directly from the volume.
24	18	S42CVLRS	4	binary	Number of tracks changed since the logical complete message was issued but before DFSMSdss read the tracks (tracks buffered in a sidefile).
28	1C	S42CVLEP	4	binary	Number of I/Os used to read data from the volume for the session.
32	20		4		Reserved.
The following relate to an EAV Concurrent Copy Session:					
0	0	S42VCID	4	binary	Logical Session ID.
4	4	S42VCRQS	3	EBCDIC	Request type, 'VCC' = Virtual Concurrent Copy.
7	7	S42VCTS	1	EBCDIC	Termination status of the session (*N* = normal termination, *A* = abnormal termination).
8	8	S42VCJNM	8	EBCDIC	Jobname that requested the CC session.
16	10	S42VCJNO	8	EBCDIC	JES number of the job that requested the CC session.
24	18	S42VCSST	8	binary	The time (TOD timestamp) that the CC session started.
32	20	S42VCEIT	8	binary	The TOD that initialization completed for the CC session.
40	28	S42VCSET	8	binary	The TOD that the concurrent copy session ended.
48	30	S42VCCTK	4	binary	Total number of tracks using CC.
52	34	S42VCVTK	4	binary	Total number of tracks using VCC.
56	38	S42VCDSP	4	binary	Total number of tracks stored in dataspaces.
60	3C	S42VCSSO	4	binary	Offset to header information for the first sybsystem in the CC session.

Table 64. Record layouts for subtype 4 (continued)

Offsets Dec.	Hex	Name	Length	Format	Description
64	40	S42VCSSN	2	binary	Number of subsystems involved in the CC session.
66	42	S42VCSSL	2	binary	Length of the SSID header.
68	44		8		Reserved.
The following relate to an EAV Concurrent Copy SSID Header Section:					
0	0	S42VSNXT	4	binary	Offset to the next SSID header (0 if last SSID).
4	4	S42VSID	2	binary	SSID.
6	6	S42VSIDP	1	binary	Storage control session ID, 0 if all extents copied using CC.
7	7		1		Reserved
8	8	S42VSMSF	4	binary	Maximum track threshold that was held in the storage control sidefile for the CC session.
12	C		8		Reserved
20	14	S42VSVLO	4	binary	Offset to first volume in the CC session for this SSID.
24	18	S42VSVLN	2	binary	Number of volumes in the CC session associated with the SSID.
26	1A	S42VSVLL	16	binary	Length of the volume section.
			4		Reserved.
The following relate to an EAV Concurrent Copy Volume section:					
0	0	S42VVLNX	4	binary	Offset to next volume for the subsystem (0 if last volume for the subsystem).
4	4	S42VVLSR	6	EBCDIC	Volume serial number for the volume.
10	A		2		Reserved.
12	C	S42VVLDV	1	binary	Device type (UCBTBYT4) for the volume.
13	D	S42VVLUA	3	EBCDIC	Unit address (UCBNAME) for the volume.
16	10	S42VVCTK	4	binary	Number of tracks processed on the volume using CC.
20	14	S42VVVTK	4	binary	Number of tracks processed on the volume using VCC.
24	18	S42VVVRD	4	binary	Number of VCC tracks read directly from DASD.
28	1C	S42VVLRD	4	binary	Number of tracks read directly from DASD.
32	20	S42VVLRS	4	binary	Number of tracks read from storage control buffers.
36	24	S42VVLEP	4	binary	Number of EXCPs.
40	28	S42VVLUX	4	EBCDIC	Unit address (UCBNAME).

Appendix C. ANTRQST and ANTRQSTL macros – call to the system data mover API

This chapter describes the application program call to the z/OS System Data Mover (SDM) application programming interface (API).

The environment, programming requirements, common invocation parameters, input and output register information are summarized, followed by the macro syntax and individual parameter descriptions. An example of the ANTRQST and ANTRQSTL macros in assembler follows the descriptions of the associated return and reason codes.

This section also contains a description of the ANTQFRVL macro (for more information, see "ANTQFRVL macro" on page 737).

Latest Updates to These Macros

Every effort has been made to provide you with the most accurate and current information for the ANTRQST and ANTRQSTL macros at the time of availability. **To stay informed about the latest functional changes and additions, refer to the extensive documentation at the top of the latest copy of these macros**. Each copy of the ANTRQST and ANTRQSTL macros contain the corresponding documentation for that release level.

This document is edited for clarity, and descriptions might not match the wording in the ANTRQST and ANTRQSTL macros.

For a information on using the REXX programming language to invoke ANTRQST, see Appendix D, "REXX support for the ANTRQST API," on page 745.

For additional information, go to DFSMS SDM Copy Services on the Web: http://www.ibm.com/systems/storage/software/sms/sdm/index.html

Table 65 shows how various z/OS functions are addressed by the ANTRQST macro.

Table 65. How the ANTRQST Macro Relates to Major Functions

ILK	Syntax	Description
ESSRVCS (FlashCopy, Global Mirror and DIAG)	"Syntax for ILK=ESSRVCS" on page 589	REQUEST=LEVEL
		REQUEST=FCESTABLISH
		REQUEST=FCQUERY
		REQUEST=FCWITHDRAW
		REQUEST=QFRVOLS
		REQUEST=QHA
		REQUEST=RSESSION
		REQUEST=RVOLUME
		REQUEST=RQUERY
		REQUEST=STATESAVE

Table 65. How the ANTRQST Macro Relates to Major Functions (continued)

ILK	Syntax	Description
PPRC (peer-to-peer remote copy)	"Syntax for ILK=PPRC" on page 596	REQUEST=FENCE
		REQUEST=LEVEL
		REQUEST=PDELPAIR
		REQUEST=PDELPATH
		REQUEST=PESTPAIR
		REQUEST=PESTPATH
		REQUEST=PFREEZE
		REQUEST=PQSCSTAT
		REQUEST=PQUERY
		REQUEST=PRECOVER
		REQUEST=PRUN
		REQUEST=PSETCHAR
		REQUEST=PSUSPEND
SNAPSHOT (Snapshot)	"Syntax for ILK=SNAPSHOT" on page 605	REQUEST=LEVEL
		REQUEST=SDVCINFO
		REQUEST=SQRYDVCS
		REQUEST=SQRYSSYS
		REQUEST=SRELEASE
		REQUEST=SSNAP
XRC (extended remote copy)	"Syntax for ILK=XRC" on page 608	REQUEST=LEVEL
		REQUEST=XADD
		REQUEST=XADVANCE
		REQUEST=XCONTIME
		REQUEST=XCOUPLE
		REQUEST=XDEL
		REQUEST=XEND
		REQUEST=XQUERY
		REQUEST=XRECOVER
		REQUEST=XSET
		REQUEST=XSCSTATS
		"Subparameters for REQUEST=XSTART" on page 729
		"Subparameters for REQUEST=XSTATUS" on page 730
		REQUEST=XSUSPEND

SDM API usability guide

This section is intended for the programmer who writes application program interfaces (APIs) that interface with the z/OS system data mover (SDM). The SDM API is intended to invoke the following Advanced Copy Services functions:

- Metro Mirror for ESS (also known as synchronous Peer-to-Peer Remote Copy - PPRC)
- Global Mirror for ESS (also known as asynchronous Peer-to-Peer Remote Copy - Asych PPRC)
- Global Copy for ESS (also known as PPRC-Extended Distance - PPRC-XD)
- Metro/Global Copy for ESS (also known as synchronous PPRC combined with PPRC-XD)
- Global Mirror for System z (also known as Extended Remote Copy - XRC)
- Metro/Global Mirror for ESS (also known as synchronous PPRC combined with Global Mirror)
- FlashCopy (FC)
- SnapShot (also known as RAMAC Virtual Array, RVA, device copy functions)

Programming requirements

The application programs that invoke Advanced Copy Services functions must meet with the following minimum requirements:

- **Authorization**

  The user identifier found in the ACEEUSRI field of the ACEE is used for all access authorization checking. This user identifier must have read access to a particular RACF FACILITY class profile.
  - For XRC requests the profile is STGADMIN.ANT.XRC.COMMANDS. In addition, for XQUERY and XSTATUS requests the profile is also STGADMIN.ANT.XRC.XQUERY.
  - For PPRC requests the profile is STGADMIN.ANT.PPRC.COMMANDS. In addition, for PQUERY requests the profile is STGADMIN.ANT.PPRC.CQUERYand for FENCE requests the profile is STGADMIN.ANT.PPRC.FENCE.
  - For SnapShot requests, the profile is STGADMIN.ANT.SNAPSHOT.COMMANDS. In addition for query requests the profile is also STGADMIN.ANT.SNAPSHOT.SQUERY.
  - For FlashCopy and Global Mirror for ESS requests the profile is STGADMIN.ANT.ESFC.COMMANDS. In addition, for FCQUERY and RQUERY requests the profile is also STGADMIN.ANT.ESFC.FCQUERY.
  - For QHA requests the profile is STGADMIN.ANT.ESS.QHA.
  - For STATESAVE requests the profile is STGADMIN.ANT.ESS.STATESAVE.
- **Dispatchable Unit Mode —** Task
- **Cross Memory Mode —** PASN=HASN=SASN
- **AMODE - 31 —** bit
- **ASC Mode —** Primary
- **Interrupt Status —** Enabled for I/O and external interrupts.
- **Locks —** No locks are held.
- **Control Parameters —** Control parameters must be in the primary address space.

For Global Mirror for System z, Metro Mirror for ESS, Global Copy for ESS, and Metro/Global Copy for ESS requests, the ANTAS000 address space must be active at the time the SDM API is invoked. For Snapshot, FlashCopy, Global Mirror for ESS, and Metro/Global Mirror for ESS requests, the ANTMAIN address space must be active at the time the SDM API is invoked.

The SDM API does not provide any data management services, such as allocation or cataloging of data sets. To copy data sets where these services are needed (such as accessing data sets on target devices), use a data set copy program such as DFSMSdss.

SDM API usage

As previously indicated, the SDM API may be used to invoke several different Advanced Copy Services functions. Even though there are several different functions that may be invoked, the SDM API macro of ANTRQST separates these functions into four different categories, each requiring a unique request type.

The ANTRQST macro must be specified with the required parameter of ILK. This parameter may be specified with one of four values. A value of ESSRVCS allows invocation of FlashCopy, Global Mirror for ESS, and Metro/Global Mirror for ESS functions. A value of PPRC allows invocation of Metro Mirror for ESS, Global Copy for ESS, and Metro/Global Copy for ESS functions. A value of SNAPSHOT allows invocation of SnapShot functions. And a value of XRC allows invocation of Global Mirror for System z functions.

In addition, the ANTRQST macro must be specified with the required parameter of REQUEST. This parameter indicates the specific type of request. The combination of an ILK value and a REQUEST value indicates the specific Advanced Copy Services function requested.

SDM API macros

As part of the z/OS component of 5695DF117, IBM ships the following macros in support of the SDM API which an application programmer needs to be aware of:

ANTRQST
: The ANTRQST macro is the macro used by the application program to invoke the SDM API. This macro then appropriately invokes the other macros required for the particular Advanced Copy Services function being requested.

  In addition, the ANTRQST macro provides the available level of SDM API support. If you specify the REQUEST=LEVEL subparameter, the ANTRQST macro returns in the RETINFO field a number indicating the level of SDM API support. Reference the prolog of the ANTRQSTL macro to determine the meaning of the number. The REQUEST=LEVEL subparameter may be specified with any ILK parameter value.

ANTFQMAP
: The ANTFQMAP macro provides the mapping of the data returned for the FlashCopy query requests (ANTRQST ILK=ESSRVCS, REQUEST=FCQUERY,FORMAT=FQMAP).

ANTPQMAP
: The ANTPQMAP macro provides the mapping of the data returned for the Metro Mirror for ESS and Global Copy for ESS query requests (ANTRQST ILK=PPRC, REQUEST=PQUERY,FORMAT=PQMAP).

ANTRQMAP
The ANTRQMAP macro provides the mapping of the data returned for the Global Mirror for ESS query requests (ANTRQST ILK=ESSRVCS, REQUEST=RQUERY,ACTION=actionvalue). The ANTRQMAP macro may be specified multiple times in your application program. The TYPE parameter of the ANTRQMAP macro has four different values which are associated with the actionvalue values that may be specified on the ACTION parameter of the ANTRQST macro. The values for the TYPE parameter are: STAT4A, STAT4B, STAT4C, and STAT51. The first 6 characters of the actionvalue value indicate the value you need to specify on the TYPE parameter of the ANTRQMAP in order to get the mapping of the data returned by the SDM API request.

ANTPQSCS
The ANTPQSCS macro provides the mapping of the data returned for Query Storage Controller Status requests (ANTRQST ILK=PPRC, REQUEST=PQSCSTAT).

ANTRQSTL
The ANTRQSTL macro provides the parameter declarations needed by the ANTRQST macro, and it also provides a listing of all the return codes returned by the SDM API. This macro must accompany the ANTRQST macro when invoking the SDM API.

In addition, the ANTRQSTL macro provides documentation of the available level of SDM API support. Reference the prolog of the ANTRQSTL macro to determine the meaning of the level number.

Note: You can specify either VERSION1 or VERSION2 for the ILK parameter on the ANTRQSTL macro. IBM recommends that you specify VERSION2, and it is the default. If you specify VERSION1 for the ILK parameter on either the ANTRQST or ANTRQSTL macro, you must specify VERSION1 for the ILK parameter on both macros.

The following macros are provided here only to indicate that their prologs will have updates which your current documentation may not reflect because of service stream updates.You will not have to program (invoke) these macros since these will be internally invoked by the ANTRQST macro based on the value you specify on the ILK parameter. You need to be aware of these macros since the prolog will reflect the latest documented support for the SDM API.

ANTRQTA
The ANTRQTA macro prolog indicates the parameters that may be specified when the ANTRQST macro is specified with ILK=XRC. The prolog also provides a description of the data returned for ILK=XRC requests.

ANTRQTB
The ANTRQTB macro prolog indicates the parameters that may be specified when the ANTRQST macro is specified with ILK=PPRC. The prolog also provides a description of the data returned for ILK=PPRC requests.

ANTRQTC
The ANTRQTC macro prolog indicates the parameters that may be specified when the ANTRQST macro is specified with ILK=SNAPSHOT. The prolog also provides a description of the data returned for ILK=SNAPSHOT requests

ANTRQTD
The ANTRQTD macro prolog indicates the parameters that may be specified when the ANTRQST macro is specified with ILK=ESSRVCS. The prolog also provides a description of the data returned for ILK=ESSRVCS requests.

All of these macros might be updated by the service stream. The documentation in this book might not reflect the latest level of support by the SDM API. Therefore, please reference the prolog of the macros for the latest documentation.

SDM API invocation

The following section describes how the ANTRQST macro is used to invoke the four different Advanced Copy Services functions. To request a particular function, you must specify the REQUEST parameter with the invocation of the ANTRQST macro. Each different value of the REQUEST parameter requires additional subparameters to be specified.

ANTRQST ILK=ESSRVCS

Specifying the ILK parameter with the ESSRVCS value on the ANTRQST macro allows invocation of the Advanced Copy Services functions of FlashCopy, Global Mirror for ESS, and Metro/Global Mirror or ESS. The following values may be specified on the REQUEST parameter to request FlashCopy, Global Mirror for ESS, and Metro/Global Mirror for ESS functions.

- The FCQUERY value specified on the REQUEST parameter allows you to determine the status of a FlashCopy relationship.
- The FCWITHDRAW value specified on the REQUEST parameter allows you to remove (withdraw) a FlashCopy relationship.
- The QFRVOLS value specified on the REQUEST parameter allows you to determine whether the source and target devices are capable and eligible for fast replication operations such as FlashCopy and Snapshot.
- The QHA value specified on the REQUEST parameter allows you to query a device and determine where path groups are established. The device is considered online where path groups are established. This might be useful, for example, when a FlashCopy or Peer-to-Peer Remote Copy establish command fails due to an online target. Data returned from QHA is mapped by the ANTQHA macro in hlq.MACLIB.
- The RQUERY value specified on the REQUEST parameter allows you to determine the status of a Global Mirror operation. The value specified on the ACTION subparameter indicates the type of status information that is being requested.
- The RSESSION value specified on the REQUEST parameter allows you to either start, stop, pause, or resume a Global Mirror session.
- The RVOLUME value specified on the REQUEST parameter allows you to add and remove volumes in a Global Mirror session.
- The STATESAVE value specified on the REQUEST parameter allows you to take a state save or a non-disruptive state save.

ANTRQST ILK=PPRC

Specifying the ILK parameter with the PPRC value on the ANTRQST macro allows invocation of the Advanced Copy Services functions of Metro Mirror for ESS, Global Copy for ESS, and Metro/Global Copy for ESS. The following values may be specified on the REQUEST parameter to request Metro Mirror for ESS, Global Copy for ESS, and Metro/Global Copy for ESS functions.

- The FENCE value specified on the REQUEST parameter allows you to prevent unintended access of a device via Soft Fence, or to prevent any hosts from bringing a device online via SPID Fence. The request can be applied to a device, a set of devices in a Logical Subsystem (LSS), or all devices in an LSS.

- The PESTPATH value specified on the REQUEST parameter allows you to establish ESCON or FCP (fibre channel protocol) paths between primary (source) site and recovery (target) site logical subsystems or storage controls. A Metro Mirror for ESS session or a Global Copy for ESS session will become active if request is successful.
- The PESTPAIR value specified on the REQUEST parameter allows you to specify a pair of primary (source) and recovery (target) volumes (devices) for active Metro Mirror for ESS session or Global Copy for ESS session.
- The PDELPAIR value specified on the REQUEST parameter allows you to remove a Metro Mirror for ESS or Global Copy for ESS volume pair from an active session.
- The PDELPATH value specified on the REQUEST parameter allows you to remove all established ESCON or FCP paths between primary site and recovery site logical subsystems or storage controls for an active Metro Mirror for ESS or Global Copy for ESS session.
- The PQUERY value specified on the REQUEST parameter allows you to determine the status of a Metro Mirror for ESS or Global Copy for ESS session and its paths and volume pairs.
- The PSUSPEND value specified on the REQUEST parameter allows you to suspend a volume pair for an active Metro Mirror for ESS or Global Copy for ESS session.
- The PFREEZE value specified on the REQUEST parameter allows you to stop all I/O operations on multiple volume pairs that have been established in an active Metro Mirror for ESS or Global Copy for ESS session whose path has been established to emulate 3990 Critical Volume mode for CGROUP automation.
- The PQSCSTAT value specified on the REQUEST parameter allows you to determine the status of the storage controller specified in the request.
- The PRUN value specified on the REQUEST parameter allows you to resume I/O operations on multiple volume pairs in an active Metro Mirror for ESS or Global Copy for ESS session whose path has been established to emulate 3990 Critical Volume mode for CGROUP automation.
- The PRECOVER value specified on the REQUEST parameter allows the recovery site to gain control of a target device on its logical subsystem or storage control. The request forces the target device into a simplex state.
- The PSETCHAR value specified on the REQUEST parameter allows you to set the characteristics of a PPRC volume pair, including whether the pair is to be used for a Preserve Mirror function in a Multi-Target Mirror configuration.

ANTRQST ILK=SNAPSHOT

Specifying the ILK parameter with the SNAPSHOT value on the ANTRQST macro allows invocation of the Advanced Copy Services function of SnapShot. The following values may be specified on the REQUEST parameter to request SnapShot functions.

- The SDVCINFO value specified on the REQUEST parameter allows you to obtain information about a RAMAC Virtual Array (RVA) device.
- The SQRYDVCS value specified on the REQUEST parameter allows you to obtain a list of RVA devices that are attached to a host.
- The SQRYSSYS value specified on the REQUEST parameter allows you to obtain a list of RVA subsystems attached to a host.

- The SSNAP value specified on the REQUEST parameter allows you to snap (quickly copy) functional tracks from one RVA device to another.
- The SRELEASE value specified on the REQUEST parameter allows you to release functional tracks on an RVA device.

ANTRQST ILK=XRC

Specifying the ILK parameter with the XRC value on the ANTRQST macro allows invocation of the Advanced Copy Services function of Global Mirror for System z. The following values may be specified on the REQUEST parameter to request Global Mirror for System z functions.

- The XSTART value specified on the REQUEST parameter allows you to start a Global Mirror for System z session.
- The XADD value specified on the REQUEST parameter allows you to add primary (source) and secondary (target) volume pairs to an active Global Mirror for System z session.
- The XCOUPLE value specified on the REQUEST parameter allows you couple together multiple Global Mirror for System z sessions to be managed by a single master session.
- The XSET value specified on the REQUEST parameter allows you to set operational and environmental values for managing Global Mirror for System z sessions.
- The XSCSTATS value specified on the REQUEST parameter allows you to obtain statistics on storage controls in a system.
- The XCONTIME value specified on the REQUEST parameter allows you to obtain the current consistency time for a Global Mirror for System z session.
- The XSUSPEND value specified on the REQUEST parameter allows you to suspend copying either volume pairs in a Global Mirror for System z session or to suspend an entire Global Mirror for System z session.
- The XADVANCE value specified on the REQUEST parameter allows you to copy data from Global Mirror for System z sessions' journals to target volumes without relabling target volumes. This function can be done for either active or suspended Global Mirror for System z sessions.
- The XRECOVER value specified on the REQUEST parameter allows you to copy data from inactive Global Mirror for System z sessions' journals to target volumes where the target volumes are relabelled (made ready to be used on the recovery site).
- The XDEL value specified on the REQUEST parameter allows you to delete (remove) volume pairs from an active Global Mirror for System z session.
- The XEND value specified on the REQUEST parameter allows you to end an active Global Mirror for System z session.

Programming considerations

The following sections will address SDM API programming considerations when programming your application program using the ANTRQST macro. This section will discuss subparameters common to all ANTRQST REQUEST values, whether an application program waits for the request to be completed or not, how data is returned to the application program, and the importance of the order different requests are made.

Subparameters common to all REQUESTs

Every ANTRQST invocation must be specified with the subparameter of RETINFO. The value specified with this subparameter references a 100 byte area used by the SDM API to return detailed information about the results of executing the request.

This area is not used to return information requested, such as when querying for information. The first word returns the return code. The second word returns the reason code. The information returned in the rest of the area depends on the REQUEST type.

The following common parameters are all optional on every ANTRQST invocation.

RETCODE
A RETCODE value may be specified to indicate the location of an area into which the SDM API will place the scheduling return code. Initial diagnostics are done by the SDM API to determine if all parameters required and specified meet the needed requirements of a request, and to determine if the environment can satisfy the request. This diagnosis is called scheduling for the purposes of this documentation. The return code is also always returned in register 15.

RSNCODE
A RSNCODE value may be specified to indicate the location of an area into which the SDM API will place the scheduling reason code. The reason code is also always returned in register 0.

PLISTVER
A PLISTVER value may be specified to indicate the SDM API parameter area size to be used by the ANTRQST macro - whether to use the smallest needed for the subparameters specified on the request, or to use the maximum allowed by the SDM API macros.

MF
A MF value may be specified to indicate the ANTRQST macro form to be used. For example, L for the list form of the macro, and E for the executible form of the macro.

Synchronizing your request

Every REQUEST value specified on the ANTRQST macro may be specified with optional subparameters that control when and how SDM responds to a request.

ASYNCH
A ASYNCH value may be specified to indicate whether the ANTRQST request is to be done asynchronously. If you specify a YES value, you must also specify the ECB subparameter to indicate the field to be posted upon completion of the request.

ECB
A ECB value must be specified if ASYNCH=YES is specified to indicate the area to be posted upon completion of a asynchronous request (as specified by ASYNCH=YES).

WAITTIME
A WAITTIME value may be specified to indicate the maximum amount of time that SDM may take before returning to the application. If the time expires before the request is completed, a return code will be sent back to the application indicating this condition, and the request may continue to be executed depending on the request. In addition, if ASYNCH=YES was also specified, the required specified ECB field will be posted.

Returned data by ANTRQST

As indicated previously, you must specify the RETINFO subparameter for every ANTRQST request, and optionally, you may specify the RETCODE and RSNCODE subparameters to get return code and reason codes placed in specific fields in addition to the RETINFO area. Register 15 (and optionally the RETCODE area) will contain the return code from scheduling the request. Register 0 (and optionally the RSNCODE area) will contain the reason code from scheduling the request.

The first eight bytes of the RETINFO area contain the execution diagnostic return and reason codes - the first four bytes contain the return code and the second four bytes contain the reason code. The information in the remaining part of the RETINFO area is dependent on the request. Reference the prolog of the appropriate macro that is used for the request, as previously mentioned in section 'SDM API Macros', to determine the information returned for the request.

If you specify the REQUEST=LEVEL subparameter, the ANTRQST macro will return a number indicating the level of SDM API support. The level number will be returned in the first four bytes of the RETINFO area. Reference the prolog of the ANTRQSTL macro to determine the meaning of the number.

Each of the four ILK types provides data for different query information requests. All of these except the ILK=XRC requests of REQUEST=XQUERY and REQUEST=XSTATUS use the subparameters of QRYINFO and QRYSIZE. These two XRC query requests use the MESSAGES subparameter value to return requested information. For all query requests, you must specify the required subparameter in order that SDM may provide the requested information.

QRYINFO

The value specified with the QRYINFO subparameter provides the area into which the SDM API will place the query information requested. SDM determines the amount of data returned based on the value specified on the QRYSIZE value specified.

QRYSIZE

The value specified on the QRYSIZE subparameter provides the size of the area indicated by the QRYINFO specified subparameter value.

MESSAGES

The ILK=XRC requests provide variable amounts of information. Therefore, the area needed to return information and messages for these requests is first determined by SDM. SDM gathers the information, and then allocates the area in the application program's address space sufficient to hold all of this information. The value specified with the MESSAGES subparameter is a four byte field into which SDM will place the address of this allocated area. For XRC requests, the information is returned in the form of XRC messages.

The information in the allocated area is provided as following format:

- Bytes 1-4 contain the size of the allocated area.
- Byte 5 contains the subpool of the allocated area.
- Byte 6 is not used.
- Bytes 7-8 contain the number of messages in the array that follows.

The rest of the allocated area is an array of messages. Each message has the following format:

- Bytes 1-2 contain the size of the message in bytes 3-x.
- Bytes 3-x contain the message.

DSNAME/DSDISP

Some of the requests (for example, REQUEST=RQUERY) allow specifying a data set name and optionally a data set disposition. The status messages will be written to the data set instead of placed in the QRYINFO or MESSAGES data area. In this situation QRYSIZE should not be specified.

Request sequence dependencies

The ANTRQST macro can be used to request many different functions. The order that these requests are made and completed is very important. In addition, determining whether your system is capable of providing all of the supported ANTRQST functions is also very important.

The first thing is issuing the ANTRQST REQUEST=LEVEL request to determine the level of support on your system. The SDM API will provide in the RETINFO area the level number indicating the current level of support on your system. See Appendix C, “ANTRQST and ANTRQSTL macros – call to the system data mover API,” on page 573 to determine the meaning of the level number and what functions are available on your system.

Once you know what functions are available on your system, you can now start invoking the ANTRQST functions. Again, the order these functions are requested and are completed is very important. For example, if you want to make ILK=PPRC requests, you must first issue the ANTRQST macro with the REQUEST=PESTPATH parameter and wait for the request to complete successfully before issuing the ANTRQST macro with the REQUEST=PESTPAIR parameter.

In other words, the SDM address spaces of ANTAS000 and ANTMAIN support multiple simultanous requests. If you were to use ASYNCH=YES and issue the REQUEST=PESTPAIR immediately after the REQUEST=PESTPATH, SDM would possibly process both of these requests at the same time, and potentially the PPRC path may not be established when the establish for the volume pair is sent to the storage control.

Programming examples

The following examples are some examples of SDM API requests that you might issue, and the data that might be returned for these requests.

ANTRQST ILK=PPRC,REQUEST=PQUERY

The following application program will issue the ILK=PPRC request of REQUEST=PQUERY to get information on the established PPRC device of X'0F70'.

```
TITLE 'APIASM01 ANTRQST TEST PROGRAM'
APIASM01 CSECT ,
APIASM01 AMODE 31
APIASM01 RMODE ANY
@MAINENT DS    0H
         USING *,15
         B     @PROLOG
         DC    AL1(10)
         DC    C' APIASM01 '
         DROP  15
@PROLOG  STM   14,12,12(13)      SAVE CALLER'S REGS IN HIS SAVEAREA
         LR    12,15
@PSTART  EQU   APIASM01
         USING @PSTART,12
*
         LR    2,1               HOLD THE INPUT PARM PTR
         GETMAIN RU,LV=DYNLEN
         LR    9,1
         USING DYNAREA,9
         LA    3,MYSA
         USING SAVEAREA,13       ADDRESS CALLER'S SAVEAREA
         ST    3,SA_NEXT         STORE MY SAVEAREA ADDRESS
         DROP  13
         USING SAVEAREA,3        ADDRESS MY SAVEAREA
         ST    13,SA_PREVIOUS    STORE CALLER'S SAVEARE ADDRESS
```

```
         DROP  3
         LR    13,3             PUT MY SAVEAREA ADDRESS IN R13
*
         ANTRQST ILK=PPRC,REQUEST=PQUERY,                              X
               DEVN=PDEVN,                                             X
               QRYSIZE=PQRYSIZE,QRYINFO=PQRYINFO,                      X
               FORMAT=PPQMAP,                                          X
               RETCODE=PRET,RSNCODE=PRSN,RETINFO=PRETINFO,             X
               MF=(E,PLIST)
*
         OPEN  (SNAPDCB,(OUTPUT))
         LA    3,SNAPDCB
         LA    4,STARTSNAP
         LA    5,ENDSNAP
         SNAP  DCB=(3),ID=1,STORAGE=((4),(5))
CLOSEIT  CLOSE SNAPDCB
*
         LR    3,13            COPY MY SAVEAREA ADDRESS
         USING SAVEAREA,3      ADDRESS MY SAVEAREA
         L     13,SA_PREVIOUS  RESTORE CALLER'S SAVEAREA ADDRESS
         DROP  3
         FREEMAIN RU,A=(9),LV=DYNLEN
         LR    0,5             RESTORE OUR RESULTS
         LR    1,6             RESTORE OUR RESULTS
         LR    15,7            RESTORE OUR RESULTS
         L     14,12(,13)      RESTORE CALLER'S RETURN ADDRESS
         LM    2,12,28(13)     RESTORE CALLER'S REGS 2-12
         BR    14              RETURN TO THE CALLER
*
*   STATIC DECLARES
*
PDEVN       DC  XL2'0F70'
PPQMAP      DC  CL6'PQMAP '
PQRYSIZE    DC  H'2400'
XDVCBLOCK   DC  CL3'WP1'
            PRINT OFF
            ANTRQSTL
SNAPDCB     DCB BLKSIZE=1632,DSORG=PS,LRECL=125,MACRF=(W),RECFM=VBA,   X
               DDNAME=SNAPDD
            PRINT ON
*
*   DYNAMIC AREA DECLARES
*
DYNLEN   EQU   DEND-DYNAREA
            ANTPQMAP
DYNAREA  DSECT
            DS    0F
MYSA        DS    18F
STARTSNAP   DS    0F
PRET        DS    F
PRSN        DS    F
PMESSAGES   DS    F
PRETINFO    DS    CL100
PQRYINFO    DS    CL2400
ENDSNAP     DS    0F
DEND     EQU  *
*
SAVEAREA DSECT
SA_RESERVED  DS F
SA_PREVIOUS  DS F
SA_NEXT      DS F
SA_R14       DS F
SA_R15       DS F
SA_R0        DS F
SA_R1        DS F
SA_R2        DS F
SA_R3        DS F
```

```
SA_R4        DS F
SA_R5        DS F
SA_R6        DS F
SA_R7        DS F
SA_R8        DS F
SA_R9        DS F
SA_R10       DS F
SA_R11       DS F
SA_R12       DS F
*
MSGAREA  DSECT
MSGSIZE  DS F
*
         END
```

The information returned by the SDM API for the ANTRQST request issued by the above application program is shown as the following output:

```
PSW AT ENTRY TO SNAP    078D1000  80007D46  ILC  02  INTC  0033
-STORAGE
0000659C                                                             00001C4D     *               ...............(*
000065A0 00000960 00000000 00000000 00000000    00000000 00000000 00000000 00000000    *...-............................*
000065C0 00000000 00000000 00000000 00000000    00000000 00000000 00000000 00000000    *................................*
      LINE 000065E0  SAME AS ABOVE
00006600 C1D5E3D7 D8D4C1D7 01010000 00000960    00000000 00000000 00000000 00000000    *ANTPQMAP.......-................*
00006620 00000000 00000000 00000000 00000000    00000000 00000000 00000000 00000000    *................................*
00006640 0F700001 CA401104 F0F0F0F0 F0F0F0F0    F1F6F9F1 00000000 00000000 00000000    *..... ..000000001691............*
00006660 CA401104 0000F0F0 F0F0F0F0 F1F6F9F1    00000000 F1F30000 00000000 00000000    *. ....0000001691....13..........*
00006680 00000000 00000000 00000000 00000000    CA800005 F1F3F0F0 F0F0F0F0 F1F6F9F1    *....................130000001691*
000066A0 00010000 00000000 02000331 00130000    00000000 00000000 00000000 00000000    *................................*
000066C0 00000000 00000000 00000000 00000000    00000000 00000000 00000000 00000000    *................................*
000066E0 00000000 00000000 CAC00006 F1F3F0F0    F0F0F0F0 F1F6F9F1 00010000 00000000    *.........{..130000001691........*
00006700 03310200 00130000 00000000 00000000    00000000 00000000 00000000 00000000    *................................*
00006720 00000000 00000000 00000000 00000000    00000000 00000000 00000000 00000000    *................................*
00006740 CA800005 00000000 00000000 00000000    00010000 00000000 03300202 00170000    *................................*
00006760 00000000 00000000 00000000 00000000    00000000 00000000 00000000 00000000    *................................*
      LINES 00006780-000067C0  SAME AS ABOVE
000067E0 00000000 00000000 00000000 00000000    00000000 00000000 00000000 0000C3A5    *..............................Cv*
00006800 00000000 00000000 00000000 00000000    00000000 00000000 00000000 00000000    *................................*
00006820 00000000 00000000 00000000 00000000    50050763 03FFC05F 00000000 00000000    *................&.....{.........*
00006840 50050763 03FFC05F 50050763 03FFC05F    50050763 04FFC441 00000000 00000000    *&.....{&.....{&.....D...........*
00006860 00000000 00000000 00000000 00000000    00000000 00000000 00000000 00000000    *................................*
      LINES 00006880-00006F40  SAME AS ABOVE
00006F60 00000000                                                                      *....                            *
```

The storage area from X'0000659C' to X'000065FF' is the RETINFO area. The first four bytes, at offset X'0000659C', indicate the execution return code of X'00001C4D' which means that a PQUERY request was processed and that the size of the QRYINFO area (indicated by the value in the second full word of X'00000960') was sufficient to hold all of the information returned for the request.

The storage area from X'00006600' to X'00006F63' is the QRYINFO area which is mapped by the ANTPQMAP area (which is indicated by the first eight bytes of the QRYINFO area).

If you use the same application program shown above, but having specified a device that was not available to the system, the SDM API returned the following information:

```
PSW AT ENTRY TO SNAP    078D1000  80007D46  ILC  02  INTC  0033
-STORAGE
0000659C                                                                 00001C48      *                .................*
000065A0 00000008 43F2F0F1 C940D5D6 40E4C3C2    40C6D6E4 D5C440C6 D6D940C4 C5E5C9C3    *.....201I NO UCB FOUND FOR DEVIC*
000065C0 C540D5E4 D4C2C5D9 40E2D7C5 C3C9C6C9    C5C440C9 D540C9D5 D7E4E340 40404040    *E NUMBER SPECIFIED IN INPUT     *
000065E0 40404040 40404040 00000000 00000000    00000000 00000000 00000000 00000000    *        ........................*
00006600 00000000 00000000 00000000 00000000    00000000 00000000 00000000 00000000    *................................*
      LINES 00006620-00006F40  SAME AS ABOVE
00006F60 00000000
```

The storage area from X'0000659C' to X'000065FF' is the RETINFO area. The first four bytes, at offset X'0000659C', indicate the execution return code of X'00001C48' which indicate that an error was encountered executing the request. The description of this return code found in ANTRQSTL indicates that byte nine of the RETINFO area (X'43') has the length of the message which starts at byte ten (which is the message text for message ANTP0201I). The error is that a UCB was not found for the device specified on the request.

Environment

These are the minimum requirements for the caller:

Minimum authorization
: The user identifier found in the ACEEUSRI field of the ACEE is used for all access authorization checking. This user identifier must have read access to a particular RACF FACILITY class profile.

  For XRC requests, the profile is STGADMIN.ANT.XRC.COMMANDS.

  For CQUERY and RQUERY requests, the profile is STGADMIN.ANT.PPRC.CQUERY.

  For all PPRC and Global Mirror requests, the profile is STGADMIN.ANT.PPRC.COMMANDS.

  For SnapShot requests, the profile is STGADMIN.ANT.SNAPSHOT.COMMANDS. FACILITY class profile.

  For non-query FlashCopy requests, the profile is STGADMIN.ANT.ESFC.COMMANDS. For query FlashCopy requests, the profile is also STGADMIN.ANT.ESFC.FCQUERY.

Dispatchable unit mode
: Task

Cross memory mode
: PASN=HASN=SASN

AMODE
: 31-bit

ASC mode
: Primary

Interrupt status
: Enabled for I/O and external interrupts

Locks
: No locks are held.

Control parameters
: Control parameters must be in the primary address space.

Programming requirements

For PPRC requests the ANTAS000 address space must be active at the time of macro invocation.

For XRC requests ANTAS000 is required to be active for the XSTART or XRECOVER commands. ANTASxxx is required to be active for all other commands at the time of macro invocation.

For Snapshot requests, the ANTMAIN address space must be active at the time of macro invocation.

For FlashCopy requests, the ANTMAIN address space must be active at the time of macro invocation.

Restrictions

Do not use the TSO or the API FlashCopy functions to copy data sets that you intend to access from the target volume. With FlashCopy, TSO or API usage does not provide any data management services, such as allocation or cataloging. Data sets that are copied using these functions are not accessible from the target volume without the user manually performing these data management tasks. You can use a data set copy program that provides these data management services as part of the copy process, such as DFSMSdss.

Recommended common invocation parameters

IBM suggests that you use the following common invocation parameters:

- For ANTRQST macro invocation, specify the following common parameters:

```
ANTRQST ILK=ilk-value,REQUEST=request,MF=(E,listname,attribute)
```

- For ANTRQSTL macro invocation with ILK=VERSION1, specify the following common parameters:

```
ANTRQSTL MF=(L, listname)
```

  Note: The specified *listname* on the ANTRQSTL macro must be the same *listname* that is used on the ANTRQST macro.
- For ANTRQSTL macro invocation with ILK=VERSION2 (the default), specify the following common parameters:

```
ANTRQSTL
NAME=parameter_list_name,              X
BASE=base_value,                       X
ILK=ilk_value,                         X
GENCODES=YES|NO,                       X
GENPLIST=YES|NO
```

Where:

NAME
: Specifies the name of the generated parameter list and the prefix of the variables used by ANTRQST. The default is PLIST.

  Example: If you code NAME=PARM_LIST, the following definition is generated.

```
PARM_LIST DS   0D
```

ANTRQST macro uses the same name parameter, NAME=PARM_LIST, thus the following definition is generated.

```
ANTRQST ...                    X
...                            X
MF=(E,PARM_LIST)
```

BASE
Specifies the alignment of the generated parameter list. The valid values are 0D and 0F. The default is 0D.

Example: If you code BASE=0F, the following definition is generated.

```
PLIST    DS    0F
```

ILK
Specifies what version of the parameter list to generate. The valid values are VERSION1 and VERSION2. The default is VERSION2.

Note: If you code VERSION1, you must also code it on the ANTRQST macro. Similarly, if you code VERSION1 for ILK on the ANTRQST macro, you must also code it on the ANTRQSTL macro.

GENCODES
Specifies whether or not to generate the definitions of the return codes. The valid values are YES and NO. The default is YES.

GENPLIST
Specifies whether or not to generate an ANTRQST parameter list. The valid values are YES and NO. The default is YES.

Input register information

Before macro invocation register 13 should point to a standard 72-byte save area.

Output register information

When control returns to the caller, the general purpose registers contain the following information:

Register	Contents
0	Reason code
1	Used as a work register by the system
2–13	Unchanged
14	Used as a work register by the system
15	Return code

Performance implications

None.

Syntax for ILK=ESSRVCS

The ANTRQST macro for ILK=ESSRVCS is written in the following format:

Parameter	Description
name	*name:* Is an optional symbol, starting in column 1, that is the name on the ANTRQST macro invocation. The name must conform to the rules for an ordinary assembler language symbol.
(blank)	One or more blanks must precede ANTRQST.
ANTRQST	
(blank)	One or more blanks must follow ANTRQST.
ILK=ESSRVCS	
REQUEST=LEVEL	
,RETINFO=*retinfo*	*retinfo:* RS-type address or address in register (2) - (12).
REQUEST=FCESTABLISH	
,SDEVN=*sdevn*	*sdevn:* RS-type address or address in register (2) - (12).
,TDEVN=*tdevn*	*tdevn:* RS-type address or address in register (2) - (12).
,DEVN=*devn*	*devno:* RS-type address or address in register (2) - (12).
,SRCSERIAL=*srcserial*	*sernum:* RS-type address or address in register (2) - (12).
,SRCSSID=*srcssid*	*ssid:* RS-type address or address in register (2) - (12).
,SRCLSS=*srclss*	*lss:* RS-type address or address in register (2) - (12).
,SRCDVC=*srcdvc*	*device:* RS-type address or address in register (2) - (12).
,TGTSERIAL=*tgtserial*	*sernum:* RS-type address or address in register (2) - (12).
,TGTLSS=*tgtlss*	*lss:* RS-type address or address in register (2) - (12).
,TGTDVC=*tgtdvc*	*device:* RS-type address or address in register (2) - (12).
,TGTUCB=*tgtucb* YES | NO	*tgtucb:* RS-type address or address in register (2) - (12). **Default:** TGTUCB=YES.
,OPENDVCS=*opendvcs* YES | NO	*device:* RS-type address or address in register (2) - (12). **Default:** OPENDVCS=NO.
,REMOTE=*remote* YES | NO	*remote:* RS-type address or address in register (2) - (12). **Default:** REMOTE=NO.
,INCREMENTAL=*incremental* NO | YES | YTW	*incremental:* RS-type address or address in register (2) - (12). **Default:** INCREMENTAL=NO.
,TGTPPRIM=*tgtpprim* YES | NO	*tgtpprim:* RS-type address or address in register (2) - (12). **Default:** TGTPPRIM=NO.
,SRCEXTENTS=*srcextents*	*srcextents:* RS-type address or address in register (2) - (12).
,TGTEXTENTS=*tgtextents*	*tgtextents:* RS-type address or address in register (2) - (12).

Parameter	Description
,RETINFO=*retinfo*	*retinfo:* RS-type address or address in register (2) - (12).
,MODE=*mode* COPY ǀ NOCOPY ǀ NO2CPY ǀ ASYNC	*mode:* RS-type address or address in register (2) - (12). **Default:** MODE=COPY.
,ONLINTGT=*onlintgt* YES ǀ NO	*onlintgt:* RS-type address or address in register (2) - (12). **Default:** ONLINTGT=NO.
,ACTION=*action* FREEZE ǀ FRR	*action*: RS-type address or address in register (2) - (12).
,SETGTOK=*setgtok* NOǀYES	*setgtok*: RS-type address or address in register (2) - (12). **Default:** SETGTOK=NO
,MSGREQ=*msgreq* YES ǀ NO	*msgreq:* RS-type address or address in register (2) - (12). **Default:** MSGREQ=NO.
,ECB=*ecb*	*ecb:* RS-type address or address in register (2) - (12). **Default:** ECB=NO_ECB.
,WAITTIME=*waittime*	*waittime:* RS-type address or address in register (2) - (12). **Default:** WAITTIME=0.
,PRESMIR=*presmir* NO ǀ REQ ǀ PREF	*presmir:* RS-type address or address in register (2) - (12). **Default:** PRESMIR=NO.
,SUBCHSET=*subchset*	*subchset:* RS-type address or address in register (2) - (12). **Default:** SUBCHSET=0.
,TSUBCHSET=*tsubchset*	*tsubchset:* RS-type address or address in register (2) - (12). **Default:** TSUBCHSET=0.
REQUEST=FCQUERY	
,DEVN=*devn*	*devn:* RS-type address or address in register (2) - (12).
,QRYSIZE=*qrysize*	*qrysize* RS-type address or address in register (2) - (12).
,QRYINFO=*qryinfo*	*qryinfo:* RS-type address or address in register (2) - (12).
,QRYSERIAL=*qryserial*	*qryserial:* RS-type address or address in register (2) - (12).
,QRYSSID=*qryssid*	*qryssid:* RS-type address or address in register (2) - (12).
,QRYLSS=*qrylss*	*qrylss:* RS-type address or address in register (2) - (12).
,OPENDVCS=*opendvcs* YES ǀ NO	*opendvcs:* RS-type address or address in register (2) - (12). **Default:** OPENDVCS=NO.
,REMOTE=*remote* YES ǀ NO	*remote:* RS-type address or address in register (2) - (12). **Default:** REMOTE=NO.
,FORMAT=*xformat* FQMAP ǀ NO	*xformat:* RS-type address or address in register (2) - (12). **Default:** FORMAT=NO.
,RETINFO=*retinfo*	*retinfo:* RS-type address or address in register (2) - (12).

Parameter	Description
,ALET=*alet*	*alet:* RS-type address or address in register (2) - (12). **Default:** ALET=0.
,ECB=*ecb*	*ecb:* RS-type address or address in register (2) - (12). **Default:** ECB=NO_ECB.
,WAITTIME=*waittime*	*waittime:* RS-type address or address in register (2) - (12). **Default:** WAITTIME=0.
,SUBCHSET=*subchset*	*subchset:* RS-type address or address in register (2) - (12). **Default:** SUBCHSET=0.
REQUEST=FCWITHDRAW	
,SDEVN=*sdevn*	*sdevn:* RS-type address or address in register (2) - (12).
,TDEVN=*tdevn*	*tdevn:* RS-type address or address in register (2) - (12).
,DEVN=*devno*	*devno:* RS-type address or address in register (2) - (12).
,SRCSERIAL=*srcserial*	*sernum:* RS-type address or address in register (2) - (12).
,SRCSSID=*srcssid*	*ssid:* RS-type address or address in register (2) - (12).
,SRCLSS=*srclss*	*lss:* RS-type address or address in register (2) - (12).
,SRCDVC=*srcdvc*	*device:* RS-type address or address in register (2) - (12).
,TGTSERIAL=*tgtserial*	*sernum:* RS-type address or address in register (2) - (12).
,TGTLSS=*tgtlss*	*lss:* RS-type address or address in register (2) - (12).
,TGTDVC=*tgtdvc*	*device:* RS-type address or address in register (2) - (12).
,TGTUCB=*tgtucb* <u>YES</u> | NO	*tgtucb:* RS-type address or address in register (2) - (12). **Default:** TGTUCB=YES.
,OPENDVCS=*opendvcs* YES | <u>NO</u>	*opendvcs:* RS-type address or address in register (2) - (12). **Default:** OPENDVCS=NO.
,REMOTE=*remote* YES | <u>NO</u>	*remote:* RS-type address or address in register (2) - (12). **Default:** REMOTE=NO.
,ACTION=*action* COMMIT | REVERT | THAW	*action:* RS-type address or address in register (2) - (12).
,RETINFO=*retinfo*	*retinfo:* RS-type address or address in register (2) - (12).
,DDSW=*ddsw* YES | <u>NO</u>	*ddsw:* RS-type address or address in register (2) - (12). **Default:** DDSW=NO.
,SRCEXTENTS=*srcextents*	*srcextents:* RS-type address or address in register (2) - (12). **Default:** SRCEXTENTS=0.
,SPACEREL=*spacerel*	*spacerel:* RS-type address or address in register (2) - (12). **Default:** SPACEREL=NO.

Parameter	Description
,TGTEXTENTS=*tgtextents*	*tgtextents:* RS-type address or address in register (2) - (12). **Default:** TGTEXTENTS=0.
,ALET=*alet*	*alet:* RS-type address or address in register (2) - (12). **Default:** ALET=0.
,ECB=*ecb*	*ecb:* RS-type address or address in register (2) - (12). **Default:** ECB=NO_ECB.
,WAITTIME=*waittime*	*waittime:* RS-type address or address in register (2) - (12). **Default:** WAITTIME=0.
,SUBCHSET=*subchset*	*subchset:* RS-type address or address in register (2) - (12). **Default:** SUBCHSET=0.
,TSUBCHSET=*tsubchset*	*tsubchset:* RS-type address or address in register (2) - (12). **Default:** TSUBCHSET=0.
REQUEST=QFRVOLS	
,CTLVOL=*ctlvol*	*ctlvol:* RS-type address or address in register (2) - (12). **Default:** CTLVOL=NO_CTLVOL.
,CTLDVC=*ctldvc*	*ctldvc:* RS-type address or address in register (2) - (12). **Default:** CTLDVC=NO_CTLDVC.
,VOLLIST=*vollist*	*vollist:* RS-type address or address in register (2) - (12).
,VOLSRCTGT=*volsrctgt*	*volsrctgt:* RS-type address or address in register (2) - (12).
,TGTPPRIM=*tgtpprim* YES | NO	*tgtpprim:* RS-type address or address in register (2) - (12). **Default:** TGTPPRIM=NO.
,RETINFO=*retinfo*	*retinfo:* RS-type address or address in register (2) - (12).
,ALET=*alet*	*alet:* RS-type address or address in register (2) - (12). **Default:** ALET=0.
, SEFLC=*seflc*, YES | NO	*seflc:* RS-type address or address in register (2) - (12). **Default:** SEFLC=NO.
,ECB=*ecb*	*ecb:* RS-type address or address in register (2) - (12). **Default:** ECB=NO_ECB.
,WAITTIME=*waittime*	*waittime:* RS-type address or address in register (2) - (12). **Default:** WAITTIME=0.
,PRESMIR=*presmir* NO | REQ | PREF	*presmir:* RS-type address or address in register (2) - (12). **Default:** PRESMIR=NO.
,RETCODE=*retcode*	*retcode:* RS-type address or address in register (2) - (12).
,RSNCODE=*rsncode*	*rsncode:* RS-type address or address in register (2) - (12).

Parameter	Description
,PLISTVER=*plistver*	*plistver:*An optional byte input decimal value in the "0–4" range that specifies the macro version. **Default:** PLISTVER=IMPLIED_VERSION.
,MF=S	**Default:** MF=S
,MF=L,*xmfctrl,xmfattr* | 0D	**Default:** MF=L,*mfctrl*,0D
,MF=M,*xmfctrl*,COMPLETE | NOCHECK	**Default:** MF=M,*mfctrl*,COMPLETE
,MF=E,*xmfctrl*,COMPLETE | NOCHECK	**Default:** MF=E,*mfctrl*,COMPLETE
REQUEST=QHA	
,DEVN=*devn*	*devn:* RS-type address or address in register (2) - (12).
,ALET=*alet*	*alet:* RS-type address or address in register (2) - (12). **Default:** ALET=0.
,CCA=*cca*	*cca:* RS-type address or address in register (2) - (12). **Default:** CCA=0.
,ECB=*ecb*	*ecb:* RS-type address or address in register (2) - (12). **Default:** ECB=NO_ECB.
,LSS=*lss*	*lss:* RS-type address or address in register (2) - (12). **Default:** LSS=0.
,QRYSIZE=*qrysize*	*qrysize* RS-type address or address in register (2) - (12).
,QRYINFO=*qryinfo*	*qryinfo:* RS-type address or address in register (2) - (12).
,RETINFO=*retinfo*	*retinfo:* RS-type address or address in register (2) - (12).
,SUBCHSET=*subchset*	*subchset:* RS-type address or address in register (2) - (12). **Default:** SUBCHSET=0.
,TGTUCB=*tgtucb* YES | NO	*tgtucb:* RS-type address or address in register (2) - (12). **Default:** TGTUCB=YES.
,WAITTIME=*waittime*	*waittime:* RS-type address or address in register (2) - (12). **Default:** WAITTIME=0.
REQUEST=RQUERY	
,SNBR=*sessno*	*sessno:* RS-type address or address in register (2) - (12).
,VOLSER=*volno*	*volno:* RS-type address or address in register (2) - (12).
,DEVN=*devno*	*devno:* RS-type address or address in register (2) - (12).
,ACTION=*action* STAT4ALSS | STAT4AESS | STAT4ACGRP | STAT4BLSS | STAT4BESS | STAT4C | STAT51 | GMPSTAT | GMLSTAT | DVCSTAT	*action:* RS-type address or address in register (2) - (12).
,LSSNBR=*lss*	*lss:* RS-type address or address in register (2) - (12).

Parameter	Description
,LSSTYPE=*lss* CKD | FB	*type:* RS-type address or address in register (2) - (12).
,DVCNBR=*nbr*	*type:* RS-type address or address in register (2) - (12).
,QRYSIZE=*size*	*size:* RS-type address or address in register (2) - (12).
,QRYINFO=*info*	*info:* RS-type address or address in register (2) - (12).
,DSNAME=*dsn*	*dsn:* RS-type address or address in register (2) - (12).
,DSDISP=*disp* OLD | MOD | SHR	*disp:* RS-type address or address in register (2) - (12).
,SUBCHSET=*subchset*	*subchset:* RS-type address or address in register (2) - (12). **Default:** SUBCHSET=0.
REQUEST=RSESSION	
,SNBR=*sessno*	*sessno:* RS-type address or address in register (2) - (12).
,VOLSER=*volno*	*volno:* RS-type address or address in register (2) - (12).
,DEVN=*devno*	*devno:* RS-type address or address in register (2) - (12).
,ACTION=*action* DEFINE | UNDEFINE | START | RESUME | PAUSE | STOP | CGPAUSE	*action:* RS-type address or address in register (2) - (12).
,LSSTYPE=*lsstype* CKD | FB	*lsstype:* RS-type address or address in register (2) - (12).
,LSSNBR=*lss*	*lss:* RS-type address or address in register (2) - (12).
,ESSSERIAL=*ess*	*ess:* RS-type address or address in register (2) - (12).
,CGINTERVAL=*cgint*	*cgint:* RS-type address or address in register (2) - (12).
,CGDRAIN=*drain*	*drain:* RS-type address or address in register (2) - (12).
,COORDINTERVAL=*coord*	*coord:* RS-type address or address in register (2) - (12).
,MSSERIAL=*msser*	*msser:* RS-type address or address in register (2) - (12).
,SBINFO=*sbinfo*	*sbinfo:* RS-type address or address in register (2) - (12).
,MASTER=*master* YES | NO	*master:* RS-type address or address in register (2) - (12).
,FORCE=*force* YES | NO	*force:* RS-type address or address in register (2) - (12).
,SUBCHSET=*subchset*	*subchset:* RS-type address or address in register (2) - (12). **Default:** SUBCHSET=0.
REQUEST=RVOLUME	
,SNBR=*sessno*	*sessno:* RS-type address or address in register (2) - (12).
,VOLSER=*serialno*	*serialno:* RS-type address or address in register (2) - (12).
,DEVN=*devno*	*devno:* RS-type address or address in register (2) - (12).

Parameter	Description
,LSSNBR=*lss*	*lss:* RS-type address or address in register (2) - (12).
,LSSTYPE=*typ* CKD ǀ FB	*typ:* RS-type address or address in register (2) - (12).
,ESSSERIAL=*ess*	*ess:* RS-type address or address in register (2) - (12).
,ACTION=*action* JOIN ǀ REMOVE	*action:* RS-type address or address in register (2) - (12).
,VOLLIST=*vlist*	*vlist:* RS-type address or address in register (2) - (12).
,VOLRANGE=*vrange*	*vrange:* RS-type address or address in register (2) - (12).
,MTVOLLIST=*vlist*	*vlist:* RS-type address or address in register (2) - (12).
,MTVOLRANGE=*vrange*	*vrange:* RS-type address or address in register (2) - (12).
,SUBCHSET=*subchset*	*subchset:* RS-type address or address in register (2) - (12). **Default:** SUBCHSET=0.
REQUEST=STATESAVE	
,DEVN=*devno*	*devno:* RS-type address or address in register (2) - (12).
,CALLER=*caller*	*caller:* RS-type address or address in register (2) - (12).
,CCA=*cca*	*caller:* RS-type address or address in register (2) - (12).
,DIAGREAS=*diagreason*	*diagreason:* RS-type address or address in register (2) - (12).
,DIAGRETC=*diagretc*	*diagretc:* RS-type address or address in register (2) - (12).
,FUNC=*function*	*function:* RS-type address or address in register (2) - (12).
,LSS=*lss*	*lss:* RS-type address or address in register (2) - (12).
,NDSS=*statesave*	*statesave:* RS-type address or address in register (2) - (12).
,SEQNO=*seqno*	*seqno:* RS-type address or address in register (2) - (12).
,SESSION=*session*	*session:* RS-type address or address in register (2) - (12).
,SUBCHSET=*subchset*	*subchset:* RS-type address or address in register (2) - (12).
,TIME=*timestamp*	*timestamp:* RS-type address or address in register (2) - (12).
,TITLE=*title*	*title:* RS-type address or address in register (2) - (12).
,TYPE=*functiontype*	*functiontype:* RS-type address or address in register (2) - (12).

Syntax for ILK=PPRC

The ANTRQST macro for ILK=PPRC is written in the following format:

Parameter	Description
name	*name:* Is an optional symbol, starting in column 1, that is the name on the ANTRQST macro invocation. The name must conform to the rules for an ordinary assembler language symbol.
(blank)	One or more blanks must precede ANTRQST.
ANTRQST	
(blank)	One or more blanks must follow ANTRQST.
REQUEST=FENCE	
,DEVN=*devn*	*devn:* RS-type address or address in register (2) - (12).
,ACTION=*action* FENCE ǀ UNFENCE ǀ SPIDFENCE ǀ SPIDUNFENCE	*action:* RS-type address or address in register (2) - (12).
,DVC=*dvc*	*dvc:* RS-type address or address in register (2) - (12).
,ECB=*ecb*	*ecb:* RS-type address or address in register (2) - (12). **Default:** ECB=NO_ECB.
,LSS=*lss*	*lss:* RS-type address or address in register (2) - (12).
,MASK=*mask*	*mask:* RS-type address or address in register (2) - (12).
,OPENDVCS=*opendvcs* YES ǀ NO	*opendvcs:* RS-type address or address in register (2) - (12). **Default:** OPENDVCS=NO.
,RETINFO=*retinfo*	*retinfo:* RS-type address or address in register (2) - (12).
,SCOPE=*scope* DEV ǀ LSS ǀ MASK	*scope:* RS-type address or address in register (2) - (12).
,SUBCHSET=*subchset*	*subchset:* RS-type address or address in register (2) - (12). **Default:** SUBCHSET=0.
,TOKEN=*token*	*waittime:* RS-type address or address in register (2) - (12).
,WAITTIME=*waittime*	*waittime:* RS-type address or address in register (2) - (12). **Default:** WAITTIME=0.
ILK=PPRC	
REQUEST=LEVEL	
,RETINFO=*retinfo*	*retinfo:* RS-type address or address in register (2) - (12).
REQUEST=PDELPAIR	
,DEVN=*devn*	*devn:* RS-type address or address in register (2) - (12).
,OPENDVCS=*opendvcs* YES ǀ NO	*opendvcs:* RS-type address or address in register (2) - (12). **Default:** OPENDVCS=NO.
,PCCA=*pcca*	*pcca:* RS-type address or address in register (2) - (12).

Parameter	Description
,PSERIAL=*pserial*	*pserial:* RS-type address or address in register (2) - (12).
,PSSID=*pssid*	*pssid:* RS-type address or address in register (2) - (12).
,SCCA=*scca*	*scca:* RS-type address or address in register (2) - (12).
,SSERIAL=*sserial*	*sserial:* RS-type address or address in register (2) - (12).
,SSSID=*sssid*	*sssid:* RS-type address or address in register (2) - (12).
,RETINFO=*retinfo*	*retinfo:* RS-type address or address in register (2) - (12).
,PLSS=*plss*	*plss:* RS-type address or address in register (2) - (12). **Default:** PLSS=NO_PLSS.
,SLSS=*slss*	*slss:* RS-type address or address in register (2) - (12). **Default:** SLSS=NO_SLSS.
,ECB=*ecb*	*ecb:* RS-type address or address in register (2) - (12). **Default:** ECB=NO_ECB.
,WAITTIME=*waittime*	*waittime:* RS-type address or address in register (2) - (12). **Default:** WAITTIME=0.
,SUBCHSET=*subchset*	*subchset:* RS-type address or address in register (2) - (12). **Default:** SUBCHSET=0.
REQUEST=PDELPATH	
,DEVN=*devn*	*devn:* RS-type address or address in register (2) - (12).
,PSERIAL=*pserialno* | ,PWWNN=*pwwnn*	*pserialno:* RS-type address or address in register (2) - (12). *pwwnn:* RS-type address or address in register (2) - (12).
,PSSID=*pssid*	*pssid:* RS-type address or address in register (2) - (12).
,SSERIAL=*sserialno* | ,SWWNN=*swwnn*	*sserial:* RS-type address or address in register (2) - (12). *swwnn:* RS-type address or address in register (2) - (12).
,SSSID=*sssid*	*sssid:* RS-type address or address in register (2) - (12).
,RETINFO=*retinfo*	*retinfo:* RS-type address or address in register (2) - (12).
,PLSS=*plss*	*plss:* RS-type address or address in register (2) - (12). **Default:** PLSS=NO_PLSS.
,SLSS=*slss*	*slss:* RS-type address or address in register (2) - (12). **Default:** SLSS=NO_SLSS.
,ECB=*ecb*	*ecb:* RS-type address or address in register (2) - (12). **Default:** ECB=NO_ECB.

Parameter	Description
,WAITTIME=*waittime*	*waittime:* RS-type address or address in register (2) - (12). **Default:** WAITTIME=0.
,SUBCHSET=*subchset*	*subchset:* RS-type address or address in register (2) - (12). **Default:** SUBCHSET=0.
REQUEST=PESTPAIR	
,DEVN=*devn*	*devn:* RS-type address or address in register (2) - (12).
,CASCADE=*cascade*	*cascade:* RS-type address or address in register (2) - (12). **Default:** CASCADE=NO.
,FORCE=*force*	*cascade:* RS-type address or address in register (2) - (12). **Default:** FORCE=NO.
,AUTORESYNC=*autoresync* YES | NO	*autoresync:* RS-type address or address in register (2) - (12). **Default:** AUTORESYNC=YES.
,INCRESYNC=*incresync*	*incresync:* RS-type address or address in register (2) - (12).
,PCCA=*pcca*	*pcca:* RS-type address or address in register (2) - (12).
,PSERIAL=*pserial*	*pserial:* RS-type address or address in register (2) - (12).
,PSSID=*pssid*	*pssid:* RS-type address or address in register (2) - (12).
,SCCA=*scca*	*scca:* RS-type address or address in register (2) - (12).
,SSERIAL=*sserial*	*sserial:* RS-type address or address in register (2) - (12).
,SSSID=*sssid*	*sssid:* RS-type address or address in register (2) - (12).
,RETINFO=*retinfo*	*retinfo:* RS-type address or address in register (2) - (12).
,PLSS=*plss*	*plss:* RS-type address or address in register (2) - (12). **Default:** PLSS=NO_PLSS.
,SLSS=*slss*	*slss:* RS-type address or address in register (2) - (12). **Default:** SLSS=NO_SLSS.
,CRIT=*crit*	*crit:* RS-type address or address in register (2) - (12). **Default:** CRIT=NO.
,OPENDVCS=*opendvcs* YES | NO	*opendvcs:* RS-type address or address in register (2) - (12). **Default:** OPENDVCS=NO.
,MODE=*mode*	*mode:* RS-type address or address in register(2) - (12). **Default:** MODE=COPY.
,MSGREQ=*msgreq*	*msgreq:* RS-type address or address in register (2) - (12). **Default:** MSGREQ=NO.
,ONLINSEC=*onlinsec*	*onlinsec:* RS-type address or address in register (2) - (12). **Default:** ONLINSEC=NO.

Parameter	Description
,PACE=*pace*	*pace:* RS-type address or address in register (2) - (12). **Default:** PACE=DEFAULT.
,ACTION=*action* FAILOVER | FAILBACK | MTFAILOVER	*action:* RS-type address or address in register (2) - (12).
,SETGTOK=*setgtok* NO | YES	*setgtok:* RS-type address or address in register (2) - (12). **Default:** SETGTOK=NO.
,ECB=*ecb*	*ecb:* RS-type address or address in register (2) - (12). **Default:** ECB=NO_ECB.
,WAITTIME=*waittime*	*waittime:* RS-type address or address in register (2) - (12). **Default:** WAITTIME=0.
,SUBCHSET=*subchset*	*subchset:* RS-type address or address in register (2) - (12). **Default:** SUBCHSET=0.
,RESETRSV=*resetrv* NO | YES	*resetrv:* RS-type address or address in register (2) - (12). **Default:** RESETRSV=NO.
REQUEST=PESTPATH	
,DEVN=*devn*	*devn:* RS-type address or address in register (2) - (12).
,LINKS=*links*	*links:* RS-type address or address in register (2) - (12). This parameter is **mutually exclusive** with the **LINK1** parameter.
,LINK1=*link1*	*link1:* RS-type address or address in register (2) - (12). This parameter is **mutually exclusive** with the **LINKS** parameter.
,LINK2=*link2*	*link2:* RS-type address or address in register (2) - (12). **Default:** LINK2=0. This parameter requires **LINK1** to be specified.
,LINK3=*link3*	*link3:* RS-type address or address in register (2) - (12). **Default:** LINK3=0. This parameter requires **LINK2** to be specified.
,LINK4=*link4*	*link4:* RS-type address or address in register (2) - (12). **Default:** LINK4=0. This parameter requires **LINK3** to be specified.
,PSERIAL=*pserialno* | ,PWWNN=*pwwnn*	*pserialno:* RS-type address or address in register (2) - (12). *pwwnn:* RS-type address or address in register (2) - (12).
,PSSID=*pssid*	*pssid:* RS-type address or address in register (2) - (12).
,SSERIAL=*sserialno* | ,SWWNN=*swwnn*	*sserialno:* RS-type address or address in register (2) - (12). *swwnn:* RS-type address or address in register (2) - (12).
,SSSID=*sssid*	*sssid:* RS-type address or address in register (2) - (12).

Parameter	Description
,PLSS=*plss*	*plss:* RS-type address or address in register (2) - (12). **Default:** PLSS=NO_PLSS.
,SLSS=*slss*	*slss:* RS-type address or address in register (2) - (12). **Default:** SLSS=NO_SLSS.
,RETINFO=*retinfo*	*retinfo:* RS-type address or address in register (2) - (12).
,CGROUP=*cgroup*	*cgroup:* RS-type address or address in register (2) - (12). **Default:** CGROUP=NO.
,RESETHP=*resethp*	*resethp:* RS-type address or address in register (2) - (12). **Default:** RESETHP=NO.
,ECB=*ecb*	*ecb:* RS-type address or address in register (2) - (12). **Default:** ECB=NO_ECB.
,WAITTIME=*waittime*	*waittime:* RS-type address or address in register (2) - (12). **Default:** WAITTIME=0.
,SUBCHSET=*subchset*	*subchset:* RS-type address or address in register (2) - (12). **Default:** SUBCHSET=0.
REQUEST=PFREEZE	
,DEVN=*devn*	*devn:* RS-type address or address in register (2) - (12).
,PSERIAL=*pserial*	*pserial:* RS-type address or address in register (2) - (12).
,PSSID=*pssid*	*pssid:* RS-type address or address in register (2) - (12).
,SSERIAL=*sserial*	*sserial:* RS-type address or address in register (2) - (12).
,SSSID=*sssid*	*sssid:* RS-type address or address in register (2) - (12).
,RETINFO=*retinfo*	*retinfo:* RS-type address or address in register (2) - (12).
,PLSS=*plss*	*plss:* RS-type address or address in register (2) - (12). **Default:** PLSS=NO_PLSS.
,SLSS=*slss*	*slss:* RS-type address or address in register (2) - (12). **Default:** SLSS=NO_SLSS.
,ECB=*ecb*	*ecb:* RS-type address or address in register (2) - (12). **Default:** ECB=NO_ECB.
,WAITTIME=*waittime*	*waittime:* RS-type address or address in register (2) - (12). **Default:** WAITTIME=0.
,SUBCHSET=*subchset*	*subchset:* RS-type address or address in register (2) - (12). **Default:** SUBCHSET=0.
REQUEST=PQSCSTAT	
,DEVN=*devn*	*devn:* RS-type address or address in register (2) - (12).
,QRYLSS=*qrylss*	*qrylss:* RS-type address or address in register (2) - (12).

Parameter	Description
,QRYSIZE=*qrysize*	*qrysize:* RS-type address or address in register (2) - (12).
,QRYINFO=*qryinfo*	*qryinfo:* RS-type address or address in register (2) - (12).
,RETINFO=*retinfo*	*retinfo:* RS-type address or address in register (2) - (12).
,OPENDVCS=*opendvcs* YES | NO	*opendvcs:* RS-type address or address in register (2) - (12). **Default:** OPENDVCS=NO.
,QRYDVC=*qrydvc*	*qrydvc:* RS-type address or address in register (2) - (12). **Default:**NO_QRYDVC.
,SCOPE=*scope* LSS | RANK	*scope:* RS-type address or address in register (2) - (12). **Default:** SCOPE=LSS.
,SUBCHSET=*subchset*	*subchset:* RS-type address or address in register (2) - (12). **Default:** SUBCHSET=0.
,ALET=*alet*	*alet:* RS-type address or address in register (2) - (12). **Default:** ALET=0.
,ECB=*ecb*	*ecb:* RS-type address or address in register (2) - (12). **Default:** ECB=NO_ECB.
,WAITTIME=*waittime*	*waittime:* RS-type address or address in register (2) - (12). **Default:** WAITTIME=0.
REQUEST=PQUERY	
,DEVN=*devn*	*devn:* RS-type address or address in register (2) - (12).
,QRYSIZE=*qrysize*	*qrysize:* RS-type address or address in register (2) - (12).
,QRYINFO=*qryinfo*	*qryinfo:* RS-type address or address in register (2) - (12).
,QRYSERIAL=*qryserial*	*qryserial:* RS-type address or address in register (2) - (12).
,QRYLSS=*qrylss*	*qrylss:* RS-type address or address in register (2) - (12).
,OPENDVCS=*opendvcs* YES | NO	*opendvcs:* RS-type address or address in register (2) - (12). **Default:** OPENDVCS=NO.
,RETINFO=*retinfo*	*retinfo:* RS-type address or address in register (2) - (12).
,FORMAT=*format* LONG | PQMAP	*format:* RS-type address or address in register (2) - (12). **Default:** FORMAT=LONG.
,ALET=*alet*	*alet:* RS-type address or address in register (2) - (12). **Default:** ALET=0.
,BITMAP=*bitmap*	*bitmap:* RS-type address or address in register (2) - (12). **Default:** BITMAP=YES.
,PATHS=*paths*	*paths:* RS-type address or address in register(2) - (12). **Default:** PATHS=NO.
,QRYSERIAL=*qryserial*	*qryserial:* RS-type address or address in register (2) - (12).

Parameter	Description
,QRYLSS=*qrylss*	*qrylss:* RS-type address or address in register (2) - (12).
,QRYDVC=*qrydvc*	*qrydvc:* RS-type address or address in register (2) - (12). **Default:**NO_QRYDVC.
,ECB=*ecb*	*ecb:* RS-type address or address in register (2) - (12). **Default:** ECB=NO_ECB.
,WAITTIME=*waittime*	*waittime:* RS-type address or address in register (2) - (12). **Default:** WAITTIME=0.
,SDEVN=*sdevn*	*sdevn:* RS-type address or address in register (2) - (12).
,SWWNN=*wwnn*	*wwnn:* RS-type address or address in register (2) - (12).
,SUBCHSET=*subchset*	*subchset:* RS-type address or address in register (2) - (12). **Default:** SUBCHSET=0.
,SSUBCHSET=*ssubchset*	*ssubchset:* RS-type address or address in register (2) - (12). **Default:** SSUBCHSET=0.
REQUEST=PRECOVER	
,DEVN=*devn*	*devn:* RS-type address or address in register (2) - (12).
,PCCA=*pcca*	*pcca:* RS-type address or address in register (2) - (12).
,PSERIAL=*pserial*	*pserial:* RS-type address or address in register (2) - (12).
,PSSID=*pssid*	*pssid:* RS-type address or address in register (2) - (12).
,SCCA=*scca*	*scca:* RS-type address or address in register (2) - (12).
,SSERIAL=*sserial*	*sserial:* RS-type address or address in register (2) - (12).
,SSSID=*sssid*	*sssid:* RS-type address or address in register (2) - (12).
,RETINFO=*retinfo*	*retinfo:* RS-type address or address in register (2) - (12).
,PLSS=*plss*	*plss:* RS-type address or address in register (2) - (12). **Default:** PLSS=NO_PLSS.
,SLSS=*slss*	*slss:* RS-type address or address in register (2) - (12). **Default:** SLSS=NO_SLSS.
,OPENDVCS=*opendvcs* YES | NO	*opendvcs:* RS-type address or address in register (2) - (12). **Default:** OPENDVCS=NO.
,OLDVOLSER=*oldvolser*	*oldvolser:* RS-type address or address in register (2) - (12). **Default:** OLDVOLSER=NO_VOLSER.
,NEWVOLSER=*newvolser*	*newvolser:* RS-type address or address in register (2) - (12). **Default:** NEWVOLSER=NO_VOLSER.
,ECB=*ecb*	*ecb:* RS-type address or address in register (2) - (12). **Default:** ECB=NO_ECB.

Parameter	Description
,WAITTIME=*waittime*	*waittime:* RS-type address or address in register (2) - (12). **Default:** WAITTIME=0.
,SUBCHSET=*subchset*	*subchset:* RS-type address or address in register (2) - (12). **Default:** SUBCHSET=0.
REQUEST=PRUN	
,DEVN=*devn*	*devn:* RS-type address or address in register (2) - (12).
,PSERIAL=*pserial*	*pserial:* RS-type address or address in register (2) - (12).
,PSSID=*pssid*	*pssid:* RS-type address or address in register (2) - (12).
,SSERIAL=*sserial*	*sserial:* RS-type address or address in register (2) - (12).
,SSSID=*sssid*	*sssid:* RS-type address or address in register (2) - (12).
,RETINFO=*retinfo*	*retinfo:* RS-type address or address in register (2) - (12).
,PLSS=*plss*	*plss:* RS-type address or address in register (2) - (12). **Default:** PLSS=NO_PLSS.
,SLSS=*slss*	*slss:* RS-type address or address in register (2) - (12). **Default:** SLSS=NO_SLSS.
,ECB=*ecb*	*ecb:* RS-type address or address in register (2) - (12). **Default:** ECB=NO_ECB.
,WAITTIME=*waittime*	*waittime:* RS-type address or address in register (2) - (12). **Default:** WAITTIME=0.
,SUBCHSET=*subchset*	*subchset:* RS-type address or address in register (2) - (12). **Default:** SUBCHSET=0.
REQUEST=PSETCHAR	
,DEVN=*devn*	*devn:* RS-type address or address in register (2) - (12).
,PCCA=*pcca*	*pcca:* RS-type address or address in register (2) - (12).
,PSERIAL=*pserial*	*pserial:* RS-type address or address in register (2) - (12).
,PLSS=*plss*	*plss:* RS-type address or address in register (2) - (12). **Default:** PLSS=NO_PLSS.
,SCCA=*scca*	*scca:* RS-type address or address in register (2) - (12).
,SSERIAL=*sserial*	*sserial:* RS-type address or address in register (2) - (12).
,SLSS=*slss*	*slss:* RS-type address or address in register (2) - (12). **Default:** SLSS=NO_SLSS.
,USEFORPM=*userforpm*	*sserial:* RS-type address or address in register (2) - (12). **Default:** USEFORPM=NO.
,RETINFO=*retinfo*	*retinfo:* RS-type address or address in register (2) - (12).

Parameter	Description
,OPENDVCS=*opendvcs* YES | NO	*opendvcs:* RS-type address or address in register (2) - (12). **Default:** OPENDVCS=NO.
,ECB=*ecb*	*ecb:* RS-type address or address in register (2) - (12). **Default:** ECB=NO_ECB.
,WAITTIME=*waittime*	*waittime:* RS-type address or address in register (2) - (12). **Default:** WAITTIME=0.
,SUBCHSET=*subchset*	*subchset:* RS-type address or address in register (2) - (12). **Default:** SUBCHSET=0.
REQUEST=PSUSPEND	
,DEVN=*devn*	*devn:* RS-type address or address in register (2) - (12).
,PRIMARY=*primary*	*primary:* RS-type address or address in register (2) - (12). **Default:** PRIMARY=NO.
,QUIESCE=*quiesce*	*quiesce:* RS-type address or address in register (2) - (12). **Default:** QUIESCE=NO.
,PCCA=*pcca*	*pcca:* RS-type address or address in register (2) - (12).
,PSERIAL=*pserial*	*pserial:* RS-type address or address in register (2) - (12).
,PSSID=*pssid*	*pssid:* RS-type address or address in register (2) - (12).
,SCCA=*scca*	*scca:* RS-type address or address in register (2) - (12).
,SSERIAL=*sserial*	*sserial:* RS-type address or address in register (2) - (12).
,SSSID=*sssid*	*sssid:* RS-type address or address in register (2) - (12).
,RETINFO=*retinfo*	*retinfo:* RS-type address or address in register (2) - (12).
,PLSS=*plss*	*plss:* RS-type address or address in register (2) - (12). **Default:** PLSS=NO_PLSS.
,SLSS=*slss*	*slss:* RS-type address or address in register (2) - (12). **Default:** SLSS=NO_SLSS.
,OPENDVCS=*opendvcs* YES | NO	*opendvcs:* RS-type address or address in register (2) - (12). **Default:** OPENDVCS=NO.
,ECB=*ecb*	*ecb:* RS-type address or address in register (2) - (12). **Default:** ECB=NO_ECB.
,WAITTIME=*waittime*	*waittime:* RS-type address or address in register (2) - (12). **Default:** WAITTIME=0.
,RETCODE=*retcode*	*retcode:* RS-type address or address in register (2) - (12).
,RSNCODE=*rsncode*	*rsncode:* RS-type address or address in register (2) - (12).

Parameter	Description
,PLISTVER=*plistver*	*plistver:*An optional byte input decimal value in the "0–4" range that specifies the macro version. **Default:** PLISTVER=IMPLIED_VERSION.
,MF=S	**Default:** MF=S
,MF=L,*mfctrl*,*mfattr* | 0D	**Default:** MF=L,*mfctrl*,0D
,MF=M,*mfctrl*,COMPLETE | NOCHECK	**Default:** MF=M,*mfctrl*,COMPLETE
,MF=E,*mfctrl*,COMPLETE | NOCHECK	**Default:** MF=E,*mfctrl*,COMPLETE
,SUBCHSET=*subchset*	*subchset:* RS-type address or address in register (2) - (12). **Default:** SUBCHSET=0.

Syntax for ILK=SNAPSHOT

The ANTRQST macro for ILK=SNAPSHOT is written in the following format:

Parameter	Description
name	*name:* Is an optional symbol, starting in column 1, that is the name on the ANTRQST macro invocation. The name must conform to the rules for an ordinary assembler language symbol.
(blank)	One or more blanks must precede ANTRQST.
ANTRQST	
(blank)	One or more blanks must follow ANTRQST.
ILK=Snapshot	
REQUEST=SDVCINFO	
,DEVN=*devn*	*devn:* RS-type address or address in register (2) - (12).
,VOLSER=*volser*	*volser:* RS-type address or address in register (2) - (12).
,UCBPTR=*ucbptr*	*ucbptr:* RS-type address or address in register (2) - (12).
,BASICSIZE=*basicsize*	*basicsize:* RS-type address or address in register (2) - (12).
,BASICINFO=*basicinfo*	*basicinfo:* RS-type address or address in register (2) - (12).
,RETINFO=*retinfo*	*retinfo:* RS-type address or address in register (2) - (12).
,ALET=*alet*	*alet:* RS-type address or address in register (2) - (12). **Default:** ALET=0.
,EXTNDSIZE=*extndsize*	*extndsize:* RS-type address or address in register (2) - (12). **Default:** EXTNDSIZE=0.
,EXTNDINFO=*extndinfo*	*extndinfo:* RS-type address or address in register (2) - (12).
,ECB=*ecb*	*ecb:* RS-type address or address in register (2) - (12). **Default:** ECB=NO_ECB.
,WAITTIME=*waittime*	*waittime:* RS-type address or address in register (2) - (12). **Default:** WAITTIME=0.

Parameter	Description
REQUEST=SQRYDVCS	
,QRYSIZE=*qrysize*	*qrysize:* RS-type address or address in register (2) - (12).
,QRYINFO=*qryinfo*	*qryinfo:* RS-type address or address in register (2) - (12).
,RETINFO=*retinfo*	*retinfo:* RS-type address or address in register (2) - (12).
,ALET=*alet*	*alet:* RS-type address or address in register (2) - (12). **Default:** ALET=0.
,SSFILTER=*ssfilter*	*ssfilter:* RS-type address or address in register (2) - (12). **Default:** SSFILTER=NO_FILTER.
,SSFSIZE=*ssfsize*	*ssfsize:* RS-type address or address in register (2) - (12).
,VLFILTER=*vlfilter*	*vlfilter:* RS-type address or address in register (2) - (12). **Default:** VLFILTER=NO_FILTER.
,VLFSIZE=*vlfsize*	*vlfsize:* RS-type address or address in register (2) - (12).
,LOWDEVN=*lowdevn*	*lowdevn:* RS-type address or address in register (2) - (12). **Default:** LOWDEVN=NO_DEVN.
,HIGHDEVN=*highdevn*	*highdevn:* RS-type address or address in register (2) - (12).
,DEVTYPE=*devtype*	*devtype:* RS-type address or address in register (2) - (12). **Default:** DEVTYPE=NO.
,PARTITION=*partition*	*partition:* RS-type address or address in register (2) - (12). **Default:** PARTITION=NO.
,ACCESS=*access*	*access:* RS-type address or address in register (2) - (12). **Default:** ACCESS=NO.
,BEGINCYL0=*begincyl0*	*begincyl0:* RS-type address or address in register (2) - (12). **Default:** BEGINCYL0=YES.
,DEVN=*devn*	*devn:* RS-type address or address in register (2) - (12). **Default:** DEVN=NO_DEVN.
,VOLSER=*volser*	*volser:* RS-type address or address in register (2) - (12).
,UCBPTR=*ucbptr*	*ucbptr:* RS-type address or address in register (2) - (12).
,ECB=*ecb*	*ecb:* RS-type address or address in register (2) - (12). **Default:** ECB=NO_ECB.
,WAITTIME=*waittime*	*waittime:* RS-type address or address in register (2) - (12). **Default:** WAITTIME=0.
REQUEST=SQRYSSYS	
,QRYSIZE=*qrysize*	*qrysize:* RS-type address or address in register (2) - (12).
,QRYINFO=*qryinfo*	*qryinfo:* RS-type address or address in register (2) - (12).
,RETINFO=*retinfo*	*retinfo:* RS-type address or address in register (2) - (12).

Parameter	Description
,ALET=*alet*	*alet:* RS-type address or address in register (2) - (12). **Default:** ALET=0.
,SSFILTER=*ssfilter*	*ssfilter:* RS-type address or address in register (2) - (12). **Default:** SSFILTER=NO_FILTER.
,SSFSIZE=*ssfsize*	*ssfsize:* RS-type address or address in register (2) - (12).
,ECB=*ecb*	*ecb:* RS-type address or address in register (2) - (12). **Default:** ECB=NO_ECB.
,WAITTIME=*waittime*	*waittime:* RS-type address or address in register (2) - (12). **Default:** WAITTIME=0.
REQUEST=SRELEASE	
,DEVN=*devn*	*devn:* RS-type address or address in register (2) - (12).
,VOLSER=*volser*	*volser:* RS-type address or address in register (2) - (12).
,UCBPTR=*ucbptr*	*ucbptr:* RS-type address or address in register (2) - (12).
,EXTINFO=*extinfo*	*extinfo:* RS-type address or address in register (2) - (12).
,EXTNBR=*extnbr*	*extnbr:* RS-type address or address in register (2) - (12).
,RETINFO=*retinfo*	*retinfo:* RS-type address or address in register (2) - (12).
,ECB=*ecb*	*ecb:* RS-type address or address in register (2) - (12). **Default:** ECB=NO_ECB.
,WAITTIME=*waittime*	*waittime:* RS-type address or address in register (2) - (12). **Default:** WAITTIME=0.
REQUEST=SSNAP	
,SDEVN=*sdevn*	*sdevn:* RS-type address or address in register (2) - (12).
,SVOLSER=*svolser*	*svolser:* RS-type address or address in register (2) - (12).
,SUCBPTR=*sucbptr*	*sucbptr:* RS-type address or address in register (2) - (12).
,SEXTINFO=*sextinfo*	*sextinfo:* RS-type address or address in register (2) - (12).
,SEXTNBR=*sextnbr*	*sextnbr:* RS-type address or address in register (2) - (12).
,TDEVN=*tdevn*	*tdevn:* RS-type address or address in register (2) - (12).
,TVOLSER=*tvolser*	*tvolser:* RS-type address or address in register (2) - (12).
,TUCBPTR=*tucbptr*	*tucbptr:* RS-type address or address in register (2) - (12).
,TEXTINFO=*textinfo*	*textinfo:* RS-type address or address in register (2) - (12).
,TEXTNBR=*textnbr*	*textnbr:* RS-type address or address in register (2) - (12).
,RETINFO=*retinfo*	*retinfo:* RS-type address or address in register (2) - (12).

Parameter	Description
,TRELOCATE=*trelocate*	*trelocate:* RS-type address or address in register (2) - (12). **Default:** TRELOCATE=NO.
,ECB=*ecb*	*ecb:* RS-type address or address in register (2) - (12). **Default:** ECB=NO_ECB.
,WAITTIME=*waittime*	*waittime:* RS-type address or address in register (2) - (12). **Default:** WAITTIME=0.
,RETCODE=*retcode*	*retcode:* RS-type address or address in register (2) - (12).
,RSNCODE=*rsncode*	*rsncode:* RS-type address or address in register (2) - (12).
,PLISTVER=*plistver*	*plistver:*An optional byte input decimal value in the "0–4" range that specifies the macro version. **Default:** PLISTVER=IMPLIED_VERSION.
,MF=S	**Default:** MF=S
,MF=L,*mfctrl*,*mfattr* ∣ 0D	**Default:** MF=L,*mfctrl*,0D
,MF=M,*mfctrl*,COMPLETE ∣ NOCHECK	**Default:** MF=M,*mfctrl*,COMPLETE
,MF=E,*mfctrl*,COMPLETE ∣ NOCHECK	**Default:** MF=E,*mfctrl*,COMPLETE

Syntax for ILK=XRC

The ANTRQST macro for ILK=XRC is written in the following format:

Parameter	Description
name	*name:* Is an optional symbol, starting in column 1, that is the name on the ANTRQST macro invocation. The name must conform to the rules for an ordinary assembler language symbol.
(blank)	One or more blanks must precede ANTRQST.
ANTRQST	
(blank)	One or more blanks must follow ANTRQST.
ILK=XRC	
REQUEST=LEVEL	
,RETINFO=*retinfo*	*retinfo:* RS-type address or address in register (2) - (12).
REQUEST=XADD	
,SID=*sid*	*sid:* RS-type address or address in register (2) - (12).
,PVOLSER=*pvolser*	*pvolser:* RS-type address or address in register (2) - (12).
,DONOTBLOCK=*xdonotblock* ∣ NO	*xdonotblock:* RS-type address or address in register (2) - (12). Mutually exclusive with DVCBLOCK parameter.
,DVCBLOCK=*xdvcblock* ∣ ON	*xdvcblock:* RS-type address or address in register (2) - (12). Mutually exclusive with DONOTBLOCK parameter.
,SVOLSER=*svolser*	*svolser:* RS-type address or address in register (2) - (12).

Parameter	Description
,COPY=*copy* | FUL | QIK	*copy:* RS-type address or address in register (2) - (12). **Default:** COPY=FUL.
,DONOTBLOCK=*donotblock* | NO | YES	*donotblock:* RS-type address or address in register (2) - (12). **Default:** DONOTBLOCK=NO.
,ERRLVL=*errlvl* | SYSTEM	*errlvl:* RS-type address or address in register (2) - (12). **Default:** ERRLVL=SYSTEM.
,SCSESSION=*scsession* | – –	*scsession:* RS-type address or address in register (2) - (12). **Default:** SCSESSION– –
,LOGPLUS=*xlogplus* | NO | YES	*xlogplus:* RS-type address or address in register (2) - (12). **Default:** LOGPLUS=NO
,SUSPENDED=*suspended*	*suspended:* RS-type address or address in register (2) - (12).
,RETINFO=*retinfo*	*retinfo:* RS-type address or address in register (2) - (12).
,MESSAGES=*messages*	*messages:* RS-type address or address in register (2) - (12).
ECB=*ecb* | NO_ECB	*ecb:* RS-type address or address in register (2) - (12). **Default:** ECB=NO_ECB.
,WAITTIME=*waittime* | 0	*waittime:* RS-type address or address in register (2) - (12). **Default:** WAITTIME=0.
REQUEST=XADVANCE	
,SID=*sid*	*sid:* RS-type address or address in register (2) - (12).
,RETINFO=*retinfo*	*retinfo:* RS-type address or address in register (2) - (12).
,MESSAGES=*messages*	*messages:* RS-type address or address in register (2) - (12).
,ECB=*ecb* | NO_ECB	*ecb:* RS-type address or address in register (2) - (12). **Default:** ECB=NO_ECB.
,ETYPE=*etype* | WAIT	*etype:* RS-type address or address in register (2) - (12). **Default:** ETYPE=WAIT.
,HLQ=*hlq* | SYS1	*hlq:* RS-type address or address in register (2) - (12). **Default:** HLQ=SYS1.
,MHLQ=*mhlq* | SYS1	*mhlq:* RS-type address or address in register (2) - (12). **Default:** MHLQ=SYS1.
,ONLINE=*online* | NO	*online:* RS-type address or address in register (2) - (12). **Default:** ONLINE=NO.
,WAITTIME=*waittime* | 0	*waittime:* RS-type address or address in register (2) - (12). **Default:** WAITTIME=0.
REQUEST=XCONTIME	
,LOGPLUS=*xlogplus* | NO | YES	*xlogplus:* RS-type address or address in register (2) - (12). **Default:** LOGPLUS=NO

Parameter	Description
,SID=*sid*	*sid:* RS-type address or address in register (2) - (12).
,RETINFO=*retinfo*	*retinfo:* RS-type address or address in register (2) - (12).
,TIME=*time*	*time:* RS-type address or address in register (2) - (12).
,ALET=*alet* ❘ 0	*alet:* RS-type address or address in register (2) - (12). **Default:** ALET=0.
,HLQ=*hlq* ❘ SYS1	*hlq:* RS-type address or address in register (2) - (12). **Default:** HLQ=SYS1.
,ECB=*ecb* ❘ NO_ECB	*ecb:* RS-type address or address in register (2) - (12). **Default:** ECB=NO_ECB.
,WAITTIME=*waittime* ❘ 0	*waittime:* RS-type address or address in register (2) - (12). **Default:** WAITTIME=0.
REQUEST=XCOUPLE	
,SID=*sid*	*sid:* RS-type address or address in register (2) - (12).
,ETYPE=*etype* ❘ ADD ❘ DELETE ❘ PURGE ❘ RELEASE	*etype:* RS-type address or address in register (2) - (12).
,RETINFO=*retinfo*	*retinfo:* RS-type address or address in register (2) - (12).
,MESSAGES=*messages*	*messages:* RS-type address or address in register (2) - (12).
,ECB=*ecb* ❘ NO_ECB	*ecb:* RS-type address or address in register (2) - (12). **Default:** ECB=NO_ECB.
,HLQ=*hlq* ❘ SYS1	*hlq:* RS-type address or address in register (2) - (12). **Default:** HLQ=SYS1.
,MHLQ=*mhlq* ❘ SYS1	*mhlq:* RS-type address or address in register (2) - (12). **Default:** MHLQ=SYS1.
,MID=*mid* ❘ NO_MID	*mid:* RS-type address or address in register (2) - (12). **Default:** MID=NO_MID.
,WAITTIME=*waittime* ❘ 0	*waittime:* RS-type address or address in register (2) - (12). **Default:** WAITTIME=0.
REQUEST=XDEL	
,SID=*sid*	*sid:* RS-type address or address in register (2) - (12).
,PVOLSER=*pvolser*	*pvolser:* RS-type address or address in register (2) - (12).
,RETINFO=*retinfo*	*retinfo:* RS-type address or address in register (2) - (12).
,MESSAGES=*messages*	*messages:* RS-type address or address in register (2) - (12).
,ECB=*ecb* ❘ NO_ECB	*ecb:* RS-type address or address in register (2) - (12). **Default:** ECB=NO_ECB.

Parameter	Description
,ATTIME=*attime* | NO_ATTIME	*attime:* RS-type address or address in register (2) - (12). **Default:** ATTIME=NO_ATTIME.
,ETYPE=*etype* | ATTIME | CANCEL | DRAIN | IMMEDIATE	*etype:* RS-type address or address in register (2) - (12). **Default:** ETYPE=IMMEDIATE.
,WAITTIME=*waittime* | 0	*waittime:* RS-type address or address in register (2) - (12). **Default:** WAITTIME=0.
REQUEST=XEND	
,SID=*sid*	*sid:* RS-type address or address in register (2) - (12).
,RETINFO=*retinfo*	*retinfo:* RS-type address or address in register (2) - (12).
,MESSAGES=*messages*	*messages:* RS-type address or address in register (2) - (12).
,ECB=*ecb* | NO_ECB	*ecb:* RS-type address or address in register (2) - (12). **Default:** ECB=NO_ECB.
,ATTIME=*attime* | NO_ATTIME	*attime:* RS-type address or address in register (2) - (12). **Default:** ATTIME=NO_ATTIME.
,ETYPE=*etype* | ATTIME | CANCEL | DRAIN | IMMEDIATE	*etype:* RS-type address or address in register (2) - (12). **Default:** ETYPE=IMMEDIATE.
,MHLQ=*mhlq* | SYS1	*mhlq:* RS-type address or address in register (2) - (12). **Default:** MHLQ=SYS1.
,WAITTIME=*waittime*	*waittime:* RS-type address or address in register (2) - (12). **Default:** WAITTIME=0.
REQUEST=XQUERY	
,SID=*sid*	*sid:* RS-type address or address in register (2) - (12).
,RETINFO=*retinfo*	*retinfo:* RS-type address or address in register (2) - (12).
,MESSAGES=*messages*	*messages:* RS-type address or address in register (2) - (12).
,ECB=*ecb* | NO_ECB	*ecb:* RS-type address or address in register (2) - (12). **Default:** ECB=NO_ECB.
,ETYPE=*etype* | CONFIGURATION | ENVIRONMENTF | ENVIRONMENTH | ENVIRONMENTP | DETAIL | DETAILS | PACE | MASTER | SET | STORAGECONTROL | SUMMARY | VOLUME| FEATURES | XFEATURES	*etype:* RS-type address or address in register (2) - (12). **Default:** ETYPE=SUMMARY.
,MHLQ=*mhlq* | SYS1	*mhlq:* RS-type address or address in register (2) - (12). **Default:** MHLQ=SYS1.
,SCSESSION=*scsession* | – –	*scsession:* RS-type address or address in register (2) - (12). **Default:** SCSESSION – –.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>,SSID=ssid</td><td>ssid: RS-type address or address in register (2) - (12).
Default: There is no specific default for this optional parameter.</td></tr>
<tr><td>,WAITTIME=waittime | 0</td><td>waittime: RS-type address or address in register (2) - (12).
Default: WAITTIME=0.</td></tr>
<tr><td>REQUEST=XRECOVER</td><td></td></tr>
<tr><td>,SID=sid</td><td>sid: RS-type address or address in register (2) - (12).</td></tr>
<tr><td>,RETINFO=retinfo</td><td>retinfo: RS-type address or address in register (2) - (12).</td></tr>
<tr><td>,MESSAGES=messages</td><td>messages: RS-type address or address in register (2) - (12).</td></tr>
<tr><td>,CHECK=check | NO</td><td>check: RS-type address or address in register (2) - (12).
Default: CHECK=NO.</td></tr>
<tr><td>,ECB=ecb | NO_ECB</td><td>ecb: RS-type address or address in register (2) - (12).
Default: ECB=NO_ECB.</td></tr>
<tr><td>,ETYPE=etype | WAIT</td><td>etype: RS-type address or address in register (2) - (12).
Default: ETYPE=WAIT.</td></tr>
<tr><td>,FORCE=force | NO</td><td>force: RS-type address or address in register (2) - (12).
Default: FORCE=NO.</td></tr>
<tr><td>,HLQ=hlq | SYS1</td><td>hlq: RS-type address or address in register (2) - (12).
Default: HLQ=SYS1.</td></tr>
<tr><td>,MHLQ=mhlq | SYS1</td><td>mhlq: RS-type address or address in register (2) - (12).
Default: MHLQ=SYS1.</td></tr>
<tr><td>,ONLINE=online | NO</td><td>online: RS-type address or address in register (2) - (12).
Default: ONLINE=NO.</td></tr>
<tr><td>,TERTIARY=tertiary | NO</td><td>tertiary: RS-type address or address in register (2) - (12).
Default: TERTIARY=NO.</td></tr>
<tr><td>,WAITTIME=waittime | 0</td><td>waittime: RS-type address or address in register (2) - (12).
Default: WAITTIME=0.</td></tr>
<tr><td>REQUEST=XSCSTATS</td><td></td></tr>
<tr><td>,QRYSIZE=qrysize</td><td>qrysize: RS-type address or address in register (2) - (12).</td></tr>
<tr><td>,RETINFO=retinfo</td><td>retinfo: RS-type address or address in register (2) - (12).</td></tr>
<tr><td>,QRYINFO=retinfo</td><td>qryinfo: RS-type address or address in register (2) - (12).</td></tr>
<tr><td>,ECB=ecb | NO_ECB</td><td>ecb: RS-type address or address in register (2) - (12).
Default: ECB=NO_ECB.</td></tr>
<tr><td>,SCSESSION=scsession | – –</td><td>scsession: RS-type address or address in register (2) - (12).
Default: SCSESSION – –.</td></tr>
</table>

Parameter	Description
,SSID=*ssid*	*ssid:* RS-type address or address in register (2) - (12). **Default:** There is no specific default for this optional parameter.
,WAITTIME=*waittime* ǀ 0	*waittime:* RS-type address or address in register (2) - (12). **Default:** WAITTIME=0.
REQUEST=XSET	
,SID=*sid*	*sid:* RS-type address or address in register (2) - (12).
,RETINFO=*retinfo*	*retinfo:* RS-type address or address in register (2) - (12).
,DVCBLOCK=*xdvcblock* ǀ ON	*xdvcblock:* RS-type address or address in register (2) - (12).
,SUSLBUSY=*syslbusy* ǀ NO ǀ YES	*syslbusy:* RS-type address or address in register (2) - (12). **Default:** SUSLBUSY=NO.
PVOLSER=*xpvolser* ǀ ALLB	*xpvolser:* RS-type address or address in register (2) - (12).
,MESSAGES=*messages*	*messages:* RS-type address or address in register (2) - (12).
,ECB=*ecb* ǀ NO_ECB	*ecb:* RS-type address or address in register (2) - (12). **Default:** ECB=NO_ECB.
COPY=*copy* ǀ FUL ǀ QIK	*copy:* RS-type address or address in register (2) - (12). **Default:** COPY=FUL.
,DVCTHLD=*dvcthld* ǀ 0 – 255	*dvcthld:* RS-type address or address in register (2) - (12). **Default:** There is no specific default.
BDVCTHLD=*bdvcthld* ǀ 0 – 255	*bdvcthld:* RS-type address or address in register (2) - (12). **Default:** There is no specific defult.
,PAGEFIX=*pagefix* ǀ 8 MB	*pagefix:* RS-type address or address in register (2) - (12). **Default:** PAGEFIX=8 MB.
,PMEMBER=*pmember* ǀ BINARY_ZEROS	*pmember:* RS-type address or address in register (2) - (12). **Default:** PMEMBER=BINARY_ZEROS.
PACTION=*paction* ǀ V ǀ A	*paction:* RS-type address or address in register (2) - (12). **Default:** PACTION=V.
PDSNAME=*pdsname* ǀ BINARY_ZEROS	*pdsname:* RS-type address or address in register (2) - (12). **Default:** PDSNAME=BINARY_ZEROS.
,PRIORITY=*priority* ǀ LOAD ǀ FIFO	*priority:* RS-type address or address in register (2) - (12). **Default:** PRIORITY=LOAD.
,RFREQUENCY=*rfrequency* ǀ 00.00.30 – 18.00.00	*rfrequency:* RS-type address or address in register (2) - (12). **Default:** RFREQUENCY=00.30.00 (minutes).
,RTRACKS=*rtracks* ǀ 0 – 99999	*rtracks:* RS-type address or address in register (2) - (12). **Default:** RTRACKS=7500.

Parameter	Description
,SCSYNCHP=*scsynchp* ∣ 0 – 45	*scsynchp:* RS-type address or address in register (2) - (12). **Default:** SCSYNCHP (primary storage control)= coordinate value with SYNCH parameter.
,SCSYNCHS=*scsynchs* ∣ 0 – 45	*scsynchs:* RS-type address or address in register (2) - (12). **Default:** SCSYNCHS (secondary storage control)= coordinate value with SYNCH parameter.
,SSID=*ssid* ∣ ALL	*ssid:* RS-type address or address in register (2) - (12). **Default:** There is no specific default for this optional parameter.
,SUSCNT=*suscnt* ∣ 0 ∣ 4096 – 63488	*suscnt:* RS-type address or address in register (2) - (12). **Default:** There is no specific default for this optional parameter.
,SUSPCT=*suspct* ∣ 0 ∣ 10 – 100	*suscnt:* RS-type address or address in register (2) - (12). **Default:** There is no specific default for this optional parameter.
,SYNCH=*synch* ∣ 0 – 45	*synch:* RS-type address or address in register (2) - (12). **Default:** SYNCH=4.
,TIMEOUT=*timeout* ∣ 00.00.01 – 18.00.00	*timeout:* RS-type address or address in register (2) - (12). **Default:** TIMEOUT=00.05.00 (minutes) - default set in most storage controls.
,UTILITY=*utility* ∣ FLT ∣ FIX	*utility:* RS-type address or address in register (2) - (12). **Default:** UTILITY=FLT (FLOAT).
,WAITTIME=*waittime* ∣ 0	*waittime:* RS-type address or address in register (2) - (12). **Default:** WAITTIME=0.
REQUEST=XSTART	
,SID=*sid*	*sid:* RS-type address or address in register (2) - (12).
,STYPE=*stype*	*stype:* RS-type address or address in register (2) - (12).
,ERRLVL=*errlvl*	*errlvl:* RS-type address or address in register (2) - (12).
,RETINFO=*retinfo*	*retinfo:* RS-type address or address in register (2) - (12).
,MESSAGES=*messages*	*messages:* RS-type address or address in register (2) - (12).
,ECB=*ecb* ∣ NO_ECB	*ecb:* RS-type address or address in register (2) - (12). **Default:** ECB=NO_ECB.
,HLQ=*hlq* ∣ SYS1	*hlq:* RS-type address or address in register (2) - (12). **Default:** HLQ=SYS1.
,MODE=*mode-list*	Intended for IBM use only.
,WAITTIME=*waittime* ∣ 0	*waittime:* RS-type address or address in register (2) - (12). **Default:** WAITTIME=0.
REQUEST=XSUSPEND	

Parameter	Description
,SID=*sid*	*sid:* RS-type address or address in register (2) - (12).
,PVOLSER=*pvolser*	*pvolser:* RS-type address or address in register (2) - (12).
,TIMEOUT=*timeout*	*timeout:* RS-type address or address in register (2) - (12).
,SCSESSION=*scsession*	*scsession:* RS-type address or address in register (2) - (12).
,DEVN=*devn*	*:* RS-type address or address in register (2) - (12).
,RETINFO=*retinfo*	*retinfo:* RS-type address or address in register (2) - (12).
,MESSAGES=*messages*	*messages:* RS-type address or address in register (2) - (12).
,ECB=*ecb* | NO_ECB	*ecb:* RS-type address or address in register (2) - (12). **Default:** ECB=NO_ECB.
,ATTIME=*attime*	*attime:* RS-type address or address in register (2) - (12). **Default:** ATTIME=NO_ATTIME.
,ETYPE=*etype* | ATTIME | CANCEL | DRAIN | IMMEDIATE	*etype:* RS-type address or address in register (2) - (12). **Default:** ETYPE=IMMEDIATE.
,MHLQ=*mhlq* | SYS1	*mhlq:* RS-type address or address in register (2) - (12). **Default:** MHLQ=SYS1.
,WAITTIME=*waittime* | 0	*waittime:* RS-type address or address in register (2) - (12). **Default:** WAITTIME=0
,RETCODE=*retcode*	*retcode:* RS-type address or address in register (2) - (12).
,RSNCODE=*rsncode*	*rsncode:* RS-type address or address in register (2) - (12).
,PLISTVER=*plistver*	*plistver:* An optional byte input decimal value in the "0–4" range that specifies the macro version. **Default:** PLISTVER=IMPLIED_VERSION.
,MF=S	**Default:** MF=S
,MF=L,*mfctrl*,*mfattr* | 0D	**Default:** MF=L,*mfctrl*,0D
,MF=M,*mfctrl*,COMPLETE | NOCHECK	**Default:** MF=M,*mfctrl*,COMPLETE
,MF=E,*mfctrl*,COMPLETE | NOCHECK	**Default: MF=E,***mfctrl***,COMPLETE**
REQUEST=XSTATUS	
,RETINFO=*retinfo*	*retinfo:* RS-type address or address in register (2) - (12).
,MESSAGES=*messages*	*messages:* RS-type address or address in register (2) - (12).
,ECB=*ecb* | NO_ECB	*ecb:* RS-type address or address in register (2) - (12). **Default:** ECB=NO_ECB.

Parameter	Description
,ETYPE=*etype* ∣ CONFIGURATION ∣ ENVIRONMENTF ∣ ENVIRONMENTH ∣ ENVIRONMENTP ∣ DETAIL ∣ DETAILS ∣ PACE ∣ MASTER ∣ SET ∣ STORAGECONTROL ∣ SUMMARY ∣ VOLUME∣ CLUSTER	*etype:* RS-type address or address in register (2) - (12). **Default:** ETYPE=CLUSTER.
,MHLQ=*mhlq* ∣ SYS1	*mhlq:* RS-type address or address in register (2) - (12). **Default:** MHLQ=SYS1.
,SCSESSION=*scsession* ∣ – –	*scsession:* RS-type address or address in register (2) - (12). **Default:** SCSESSION – –.
,SSID=*ssid*	*ssid:* RS-type address or address in register (2) - (12). **Default:** There is no specific default for this optional parameter.
,WAITTIME=*waittime* ∣ 0	*waittime:* RS-type address or address in register (2) - (12). **Default:** WAITTIME=0.

Parameter descriptions for ILK=ESSRVCS

When you invoke the ANTRQST macro for FlashCopy or Global Mirror, you can use the following REQUEST parameters. For a list of the Common End Codes associated with each request within this ILK, see "Common end codes for REQUESTS within each ILK" on page 735.

Subparameters for REQUEST=LEVEL

REQUEST=LEVEL

Asks for the level of ANTRQST that is installed on the system.

,RETINFO=*retinfo*

is the name (RS-type) (or address in register (2)-(12) ASM only) of a required 100-character output field used to return the level of ANTRQST installed on the system. If ANTRQST is not installed, return code 7000 (RQST_PC_NUMBER_ZERO - see ANTRQSTL) is returned in register 15 and placed in the RETCODE area, if specified. If ANTRQST is installed, a level value is placed in the first 4-bytes of the RETINFO field. This level value indicates which ANTRQST functions, keywords and parameters are supported. For more information about the LEVEL request output, see the prolog comments in the ANTRQSTL macro (shipped with the ANTRQST macro).

Subparameters for REQUEST=FCESTABLISH

REQUEST=FCESTABLISH

Asks for the Establish FlashCopy relationship function to be executed.

,SDEVN=*sdevn*

Specifies the 2-byte hexadecimal device number of the source device. If the field contains the null value (X'0000'), the parameter is treated as omitted. Device number X'0000' cannot be used with the FlashCopy FCESTABLISH request.

For FlashCopy, the source device can be the same as the target device. In this case, the source and target extents must be specified, and the extents cannot overlap.

For Open System (fixed block) devices, this parameter is invalid. If specified with fixed block devices, this parameter must be set to the null (X'0000') value.

To code: Specify the RS-type address, or address in register (2)-(12), of an required 2-character input field.

,TDEVN=*tdevn*

Specifies the 2-byte hexadecimal device number of the target device. If the field contains the null value (X'0000'), the parameter is treated as omitted. Device number X'0000' cannot be used with the FlashCopy establish request.

For FlashCopy, the target device can be the same as the source device. In this case, the source and target extents must be specified, and the extents cannot overlap.

For Open System (fixed block) devices, this parameter is invalid. If specified with fixed block devices, this parameter must be set to the null (X'0000') value.

To code: Specify the RS-type address, or address in register (2)-(12), of an required 2-character input field.

,DEVN=*devn*

Specifies the binary device number to use for I/O.

When OPENDVCS(YES) is specified for a local FlashCopy relationship (REMOTE(NO) is specified or allowed to default), DEVN must specify a CKD access volume located in the same subsystem cluster as the fixed block device identified by SOURCE in this command.

When OPENDVCS(YES) is specified with (REMOTE(YES), DEVN must specify a CKD access volume located in the same subsystem cluster as the PPRC primary device that is paired with the PPRC secondary specified as the FlashCopy source.

When REMOTE(YES) is specified for CKD volumes (OPENDVCS(NO) is specified or allowed to default), DEVN must specify the PPRC primary device that is paired with the PPRC secondary specified as the FlashCopy source.

DEVN must be a 4-digit hexadecimal address of a configured device with a UCB on the System z system issuing the command.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 2-character field.

,SRCSERIAL=*srcserial*

When OPENDVCS(YES) or REMOTE(YES) is specified, SRCSERIAL identifies the storage control serial number that can include up to 10 digits, depending on the type of storage control.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 10-character field.

,SRCLSS=*srclss*
When OPENDVCS=YES or REMOTE=YES is specified, this identifies the two-digit hexadecimal value for the logical subsystem (LSS) for the device (ESS only).

Note: The LSS number is required if the storage control supports logical subsystems (like the ESS) and not allowed if the storage control does not support logical subsystems.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 1-character field.

,SRCDVC=*srcdvc*
When OPENDVCS=YES or REMOTE=YES is specified, this identifies the two-digit hexadecimal value for the device (either logical unit number or channel connection address).

To code: Specify the RS-type address, or address in register (2)-(12), of a required 1-character field.

,TGTSERIAL=*tgtserial*
When OPENDVCS=YES or REMOTE=YES is specified, TGTSERIAL identifies the storage control serial number that can include up to 10, depending on the type of storage control.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 10-character field.

,TGTLSS=*tgtlss*
When OPENDVCS=YES or REMOTE=YES is specified, this identifies the two-digit hexadecimal value for the logical subsystem (LSS) for the device (ESS only).

Note: The LSS number is required if the storage control supports logical subsystems (like the ESS) and not allowed if the storage control does not support logical subsystems.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 1-character field.

,TGTDVC=*tgtdvc*
When OPENDVCS=YES or REMOTE=YES is specified, this identifies the two-digit hexadecimal value for the device (either logical unit number or channel connection address).

To code: Specify the RS-type address, or address in register (2)-(12), of a required 1-character field.

,TGTUCB=*tgtucb*
,TGTUCB=YES
Specifies YES or NO, indicating whether an MVS device number will be used for target addressability. Yes is the default.

YES
Indicates that the SDEVN and TDEVN keywords are being used to identify the source and target devices in the relationship, using MVS device numbers.

NO Indicates that an MVS device number will not be used for the target device (TDEVN keyword will not be used). Instead, the SDEVN, TGTSERIAL, TGTLSS, and TGTDVC keywords will be used.

Note: TGTUCB(NO) is not supported when OPENDVCS(YES) or REMOTE(YES) is specified.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 3-character field.

,SRCSSID=*srcssid*
When REMOTE=YES is specified without OPENDVCS=YES, this is the two-byte binary unsigned SSID of the subsystem where the FlashCopy request is to occur. This must be the same value as that specified for SSID on the PPRC secondary volume on the PPRC establish pair command.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional half-word field.

,OPENDVCS=*opendvcs*
,OPENDVCS=NO
Specifies whether Open System (fixed block) devices or CKD devices are addressed by this command. The values are:

NO The required parameters (SDEVN and TDEVN) identify CKD devices. The default is NO.

YES
Specifies that the required parameters (DEVN, SRCSERIAL, SRCLSS, SRCSSID, SRCDVC, TGTSERIAL, TGTLSS, TGTDVC) identify fixed block devices using a CKD access device.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 3-character field.

,REMOTE=*remote*
,REMOTE=NO
Specifies whether the request is for a subsystem that is not directly attached to the issuing processor. The values are:

NO The request is directed to a device on a subsystem locally attached to the issuing processor. The default is NO.

YES
The request is directed to a remote subsystem using Inband subsystem functions. When REMOTE=YES is specified, the request must be for a full volume.

When specifying an Inband request for a CKD secondary device, DEVN must identify a PPRC primary device in a subsystem accessible by the host processor issuing the request, and the FlashCopy source device must be the PPRC secondary of that DEVN.

When specifying an Inband request for an Open System (fixed block) secondary device, DEVN must identify an online CKD System z device located in the same subsystem cluster as the fixed block device of the PPRC primary paired with the secondary designated as FlashCopy source.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 3-character field.

,INCREMENTAL=*incremental*
,INCREMENTAL=NO
Specifies whether the FlashCopy establish relationship remains active after

initial copy is complete allowing subsequent changes to be tracked so that future FlashCopy operations require only a subset of the volume to be copied. The values are:

NO The FlashCopy relationship ends after the background copy has completed (when MODE=COPY) is specified) or all source and target tracks have been updated (when MODE=NOCOPY) is specified). The default is NO.

YES
The FlashCopy relationship remains in effect after the request completes. Subsequent changes are tracked so that future FlashCopy operations are performed incrementally. This relationship continues until explicitly terminated with a FlashCopy Withdraw request.

YTW
The FlashCopy relationship remains in effect after the request completes. Subsequent changes are tracked so that future FlashCopy operations are performed incrementally. This relationship continues until explicitly terminated with a FlashCopy Withdraw request. YTW performs the same function as INCREMENTAL=YES except that YTW allows the target to be writable.

Note: The FlashCopy target is writable while the incremental relationship is active. Any writes done to the target during this period are overwritten if a subsequent increment is done, keeping the target a true copy of the source. If the relationship is reversed, the changes made to the target are reflected on the source.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 3-character field.

,TGTPPRIM=*tgtpprim*
,TGTPPRIM=NO
Specifies whether the target in the FlashCopy relationship can be the primary in a PPRC pair. The values are:

NO The target in this FlashCopy relationship cannot be the primary in a PPRC pair. The default is NO.

YES
The target in this FlashCopy relationship can be the primary in a PPRC pair. This request proceeds normally to the specified target but the hardware ignores the PPRC status of the target.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 3-character field.

,SRCEXTENTS=*srcextents*
Is the extent data structure described below. The extents in the extent data structure describe the tracks on the source CKD volume that is active in this FlashCopy relationship.

The structure of the extent data is as follows:

- A 4-byte field with a hexadecimal count of the source extent fields in the following list. This count field must have a value of 1 to 110 for ESS subsystems that have volume FlashCopy installed. This count field must have a value of 1 to the maximum allowable number of extents for a device in ESS subsystems that have FlashCopy installed.

This field can also contain a 4-character value of ALL. If this value is present, all tracks of the source device are active in the FlashCopy relationship (full volume). If this value is present, no other source extent information is used, and target extent information is ignored.

- A 4-byte reserved field.
- A list of source extent fields. Each 8-byte extent field has two extents, the beginning source extent and the ending source extent on the volume to be part of this FlashCopy relationship.

  The extent format is CCHH where CC is an unsigned 16-bit binary cylinder number. HH is an unsigned 16-bit binary track number.

  The extents in each source extent field contain the following:
  - A 4-byte beginning source extent.
  - A 4-byte ending source extent.

  Each source extent field must have a related target extent field identifying the same number of tracks.

  The tracks identified in the source extent fields must be valid for the source device.

To code: Specify the RS-type address, or address in register (2)-(12) (ASM only), of a required variable character input field, aligned on a fullword.

,TGTEXTENTS=*tgtextents*

Is the extent data structure described below. The extents in the extent data structure describe the tracks on the target volume that will be active in this FlashCopy relationship.

The structure of the extent data is as follows:

- A 4-byte field with a binary count of the target extent fields in the following list. This count field must have a value of 1 to 110.

  This field can also contain a 4-character value of ALL. This indicates that this establish is a full volume request.
- A 4-byte reserved field.
- A list of target extent fields. Each 8-byte extent field has two extents, the beginning extent and the ending extent on the target volume to be part of this FlashCopy relationship.

  The extent format is CCHH where CC is an unsigned 16-bit binary cylinder number. HH is an unsigned 16-bit binary track number.

  The extents in each target extent field contain the following format:
  - A 4-byte beginning target extent.
  - A 4-byte ending target extent.

To code: Specify the RS-type address, or address in register (2)-(12), of a required variable character input field, aligned on a fullword.

,RETINFO=*retinfo*

Is the name of a required 100-character output field that is used to return detailed information about the results of **executing** the request. The output field is a name (RS-type) or an address in register (2)-(12). The program returns information about the results of **scheduling** the request in the RETCODE and RSNCODE fields. The area should align on a word boundary. The first 4-bytes contain the return code, and the second 4-bytes contain the reason code. The remainder of the information is dependent on the return and reason codes. See the coding example about how to coordinate the RETCODE, RSNCODE, and RETINFO fields.

,MODE=*mode*
,MODE=COPY

Specifies the type of FlashCopy relationship to be started for this pair of volumes.

COPY requests the program to establish a FlashCopy relationship between the source device and the target device. The program starts a background copy of all tracks within the specified extents from the source volume to the target volume. When the background copy completes, the FlashCopy relationship terminates. The target volume extents contain the same data as the source volume extents when the FlashCopy relationship was first established.

NOCOPY requests the program to establish a FlashCopy relationship between the source device and the target device. The program does not do a background copy of tracks from source volume to target volume. Processing of data on the source volume is the same as if the source volume was not in a FlashCopy relationship. Records read from the target volume within the specified extents will have the same data as the related source volume records at the time the FlashCopy relationship was established.

Note: When a FlashCopy NOCOPY relationship is ended, the track data on the target device is unpredictable and should not be used. If updates occur to source device tracks in the FlashCopy NOCOPY relationship, a copy of the source tracks from the point-in-time of the FlashCopy establish may or may not be written to the target device.

Tracks may be copied from the source to the target volume even if the source track is not changed. This includes the track that contains the volume label. Therefore, to avoid duplicate volume serial problems when the target device is later varied online, IBM recommends that you relabel the target volume after withdrawing a volume-level FlashCopy NOCOPY relationship.

NO2CPY initiates a background copy from the source to target. When a MODE=NOCOPY relationship already exists between source and target, this relationship ends when the background copy is completed. There must be an existing FlashCopy relationship between source and target. If one does not exist, none will be created.

ASYNC indicates that this is being established to a volume set that is currently, or will be part of a Global Mirror for ESS session.

The default is COPY.

To code: Specify the RS-type name, or address in register (2)-(12), of an optional 6-character input field. The value is left-justified and padded on the right with blanks. If the field contains binary zeros, the program uses the default value.

,ONLINTGT=*onlintgt*
,ONLINTGT=NO

Is an optional 3-character input field that contains a keyword. This keyword specifies whether the FlashCopy Establish should continue if the specified target device is in an online state to any system. The name of the input field is a name (RS-type) or address in register (2)-(12). The value is left-justified and padded on the right with blanks. If the field contains binary zeros, the program uses the default.

NO indicates fail the FlashCopy Establish if the target device is online to any system.

YES indicates continue the FlashCopy Establish if the target device is online to any system.

The default is NO.

ACTION=*action*

Identifies that an action is to take place during the FlashCopy establish request.

FREEZE

Specifies that the FlashCopy source volume is to be part of a FlashCopy consistency group. The FlashCopy relationship is established between the source and target volumes, or extents, and all I/O to the source volume will be held (results in a long busy) until one of the following conditions is met:

- A FlashCopy withdraw with action THAW is processed by the LSS where the volume resides.
- A two-minute timer has expired. (The two-minute time can be adjusted using the ESS Specialist GUI.)

ACTION=FREEZE is mutually exclusive with MODE(ASYNC).

FRR

Specifies Fast Reverse Restore, which is a function to be used with Global Mirror or Metro/Global Mirror for ESS when recovering from an outage. This reverses the direction of the FlashCopy relationship, restoring the source volume to the state it was in when it last flashed to the target. Changed tracks are copied back from the target to the source.

To code: Specify the RS-type name, or address in register (2)-(12), of an optional 12-byte (character) field. The value is left-justified and padded on the right with blanks.

SETGTOK=*setgtok*
,SETGTOK=NO

Specifies whether the target of the specified full volume relationship can be a space-efficient volume.

NO indicates that the target cannot be a space-efficient volume. The default is NO.

YES indicates that the target can be a space-efficient volume. If an out of space condition occurs, the relationship is failed.

To code: Specify the RS-type name, or address in register (2)-(12), of a 3-character field. The specified character value is left-justified and padded on the right with blanks.

,MSGREQ=*msgreq*
,MSGREQ=NO

Specifies whether to wait for FlashCopy Establish initialization to complete.

NO indicates to not wait.

YES indicates to wait. Only valid when MODE is COPY.

The default is NO.

To code: Specify the RS-type name, or address in register (2)-(12), of an optional 3-character field. The specified character value is left-justified and padded on the right with blanks. If the field contains binary zeros, the program uses the default.

,ECB=*ecb*
,ECB=NO_ECB
Is an optional fullword input field that SDM will post for an asynchronous request. The name of the input field is a name (RS-type) or address in register (2)-(12). For synchronous requests, the program ignores this field.

The default is NO_ECB.

,WAITTIME=*waittime* **| 0**
Specifies how long in seconds SDM will wait for a request to complete. A value of zero says to not time the request. If the time expires before the request is complete, the program returns a value of 7039 (RQST_WAITTIME_EXPIRED). For more information about RQST_WAITTIME_EXPIRED, refer to ANTRQSTL. If the request is a synchronous request, the program returns the value in the return code part of RETINFO. If the request is an asynchronous request, the program uses the value as the ECB post code.

The default is 0.

To code: Specify the RS-type name, or address in register (2)-(12), of an optional halfword input field. The specified character value is left-justified and padded on the right with blanks. A value of zero says to not time the request.

,PRESMIR=*presmir*
,PRESMIR=NO
Specifies the handling of the request based on whether the specified target is a PPRC primary device. The values are:

NO indicates that the FCESTABLISH request is to be performed without considering a Preserve Mirror operation.

REQ indicates that if the specified target device is a Metro Mirror primary device, the pair must not go into a duplex pending state as the result of this FCESTABLISH request.

PREF indicates that if the specified target device is a Metro Mirror primary device, it would be preferable that the pair not go into a duplex pending state as the result of the FCESTABLISH request. However, if the duplex pending state cannot be avoided, the FCESTABLISH request should still be performed.

The default is NO.

To code: Specify the RS-type name, or address in register (2)-(12), of an optional 4-character input field. The specified character value is left-justified and padded on the right with blanks. A value of zero says to not time the request.

,SUBCHSET=*subchset*
,SUBCHSET=0
Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is currently supported by the host system's processor and configured for the device.

If you specify a value of X'00' or omit this keyword, the command is issued to the device that is currently logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

To code: Specify the RS-type name, or address in register (2)-(12), of an optional 1-byte (character) field.

,TSUBCHSET=*tsubchset*
,TSUBCHSET=0

Used with the TDEVN parameter. *tsubchset* is the name (RS-type) or address in register (2)-(12) of a 1-character field that specifies the subchannel set is to be used to get information about the target device specified with the TDEVN parameter. This is the subchannel set for the device as defined in the Hardware Configuration Dialog (HCD). The values are determined by what is currently supported by the host system's processor and configured for the device.

If you specify a value of X'00' or omit this keyword, the command is issued to the device that is currently logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

Subparameters for REQUEST=FCQUERY

REQUEST=FCQUERY

Asks for the Query FlashCopy relationship function to be executed.

,DEVN=*devn*

A required input parameter field containing the binary device number to use for I/O. DEVN must be a 4-digit hexadecimal address of a configured device with a UCB on the System z system issuing the command.

When querying a local CKD device, this specifies the four-digit hexadecimal device number of the volume.

When querying a local fixed block device (OPENDVCS(YES) is specified), DEVN must specify a CKD access volume located in the same subsystem cluster as the FB device identified by QRYDVC in this command.

When querying a remote fixed block device (both OPENDVCS(YES) and REMOTE(YES) are specified), DEVN must specify a CKD access volume located in the same subsystem cluster as the PPRC primary device that is paired with the PPRC secondary acting as the FlashCopy source.

When querying a remote CKD device (REMOTE(YES) is specified), DEVN must specify the PPRC primary device that is paired with the PPRC secondary acting as the FlashCopy source.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-byte field.

QRYDVC =*qrydvc*

When OPENDVCS=YES or REMOTE=YES is specified, this specifies the two-digit hexadecimal value for the device being queried (either connection address or logical unit number).

To code: Specify the RS-type address, or address in register (2)-(12), of a required 1-character field.

,QRYSIZE=*qrysize*

specifies the length of the QRYINFO area.

To code: Specify the RS-type address, or address in register (2)-(12), of a required halfword input field.

,QRYSERIAL=*qryserial*
When OPENDVCS=YES or REMOTE=YES is specified, this identifies the storage control serial number that can include up to 10 digits, depending on the type of storage control.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 10-character field.

,QRYLSS=*qrylss*
When OPENDVCS=YES or REMOTE=YES is specified, this identifies the two-digit hexadecimal value for the logical subsystem (LSS) for the device (ESS only).

Note: The LSS number is required if the storage control supports logical subsystems (like the ESS) and not allowed if the storage control does not support logical subsystems.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 1-character field.

,QRYSSID=*qryssid*
When REMOTE=YES is specified without OPENDVCS=YES, this is the two-byte binary unsigned SSID of the subsystem where the FlashCopy request is to occur. This must be the same value as that specified for SSID on the PPRC secondary in the PPRC establish command.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional half-word field.

,OPENDVCS=*opendvcs*
,OPENDVCS=NO
Specifies whether Open System (fixed block) devices or CKD devices are addressed by this command. The values are:

NO Specifies that the required parameters identify a CKD device. The default is NO.

YES
Specifies that the required parameters identify a fixed block device.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 3-character field.

,REMOTE=*remote*
,REMOTE=NO
Specifies whether the request is for a subsystem that is not directly attached to the issuing processor. The values are:

NO The request is directed to a device on a subsystem locally attached to the issuing processor. The default is NO.

YES
The request is directed to a remote subsystem using Inband subsystem functions. When REMOTE=YES is specified, the request must be for a full volume.

When specifying an Inband request for a CKD secondary device, DEVN must identify a PPRC primary device in a subsystem accessible by the host processor issuing the request, and the FlashCopy source device must be the PPRC secondary of that DEVN.

When specifying an Inband request for an Open System (fixed block) secondary device, DEVN must identify an online CKD System z device located in the same subsystem cluster as the fixed block device of the PPRC pair with the secondary designated as FlashCopy target.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 3-character field.

,FORMAT=FQMAP | NO
specifies whether the data is to be formatted using the ANTFQMAP macro. Specify the value FQMAP left-justified and padded on the right with blanks for formatting. A value of binary zeros defaults to NO.

To code: Specify the RS-type name, or address in register (2)-(12), of an optional 6-character input field.

,QRYINFO=*qryinfo*
specifies the name of a variable-character output field that returns the result of the query.

If the field is not big enough to contain the complete result, the program places a return code of 7622 (RQST_FCQUERY_QRYINFO_TOO_SMALL) in the return code part of RETINFO. The program places the total space required in the reason code part of RETINFO.

If the field is big enough to contain the complete result, the program places a return code of 7623 (RQST_FCQUERY_QRYINFO_LARGE_ENOUGH) in the return code part of RETINFO. The program places the total amount of space that the result uses in the reason code part of RETINFO.

You can use the optional ALET parameter to specify the location of the QRYINFO field.

To code: Specify the RS-type name, or address in register (2)-(12), of a required variable-character output field.

,RETINFO=*retinfo*
Is a required 100-character output field that the program uses to return detailed information about the results of **executing** the request. The name of the output field is a name (RS-type) or address in register (2)-(12). The program returns information about the results of **scheduling** the request in the RETCODE and RSNCODE fields. Align the area on a word boundary. The first 4 bytes contain the return code. The second 4 bytes contain the reason code. The remainder of the information depends on the return and reason codes. See the coding example about how to coordinate the RETCODE, RSNCODE, and RETINFO fields.

,ALET=*alet* **| 0**
Is an optional fullword input field that specifies the ALET value that the program uses for the QRYINFO field, the SRCEXTENTS field, and the TGTEXTENTS field. The name of the input field is a name (RS-type) or address in register (2)-(12). If you specify the ALET parameter, the program uses the ALET value to reference above-listed fields. Otherwise, the program does not use the ALET value to reference any of the output fields listed above.

The default is 0.

,ECB=*ecb*

,ECB=NO_ECB
Is an optional fullword input field that SDM posts for an asynchronous request. The name of the input field is a name (RS-type) or address in register (2)-(12). The program ignores this field for synchronous requests.

The default is NO_ECB.

,WAITTIME=*waittime* | 0
Is an optional halfword input field that contains how long in seconds SDM waits for a request to complete. The name of the input field is a name (RS-type) or address in register (2)-(12). A value of zero specifies that the requests should not be timed. If the time expires before the request completes, the program returns a value of 7039 (RQST_WAITTIME_EXPIRED). For information about RQST_WAITTIME_EXPIRED, see ANTRQSTL. If the request is a synchronous request, the program places the value in the return code part of RETINFO. If the request is an asynchronous request, the program uses the value as the ECB post code.

The default is 0.

,SUBCHSET=*subchset*
,SUBCHSET=0
Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is currently supported by the host system's processor and configured for the device.

If you specify a value of X'00' or omit this keyword, the command is issued to the device that is currently logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

To code: Specify the RS-type name, or address in register (2)-(12), of an optional 1-byte (character) field.

Subparameters for REQUEST=FCWITHDRAW

REQUEST=FCWITHDRAW
Asks for the Withdraw FlashCopy relationship function to be executed.

,SDEVN=*sdevn*
specifies the 2-byte device number of the source device.

For Open System (fixed block) devices, this parameter is invalid. If specified with fixed block devices, this parameter must be set to the null (X'0000') value.

To code: Specify the RS-type name, or address in register (2)-(12), of an optional 2-character input field.

,TDEVN=*tdevn*
specifies the binary device number of the target device.

For Open System (fixed block) devices, this parameter is invalid. If specified with fixed block devices, this parameter must be set to the null (X'0000') value.

To code: Specify the RS-type name, or address in register (2)-(12), of a required 2-byte input field.

,DEVN=*devn*

A required input parameter field containing the binary device number to use for I/O. DEVN must be a 4-digit hexadecimal address of a configured device with a UCB on the System z system issuing the command.

When OPENDVCS=YES is specified, DEVN identifies a CKD access volume located in the same subsystem cluster as the open volume specified as SOURCE in this command.

When OPENDVCS=YES and REMOTE=YES are both specified, DEVN identifies a CKD access volume located in the same subsystem cluster as the PPRC primary paired with the PPRC secondary acting as the FlashCopy source.

When REMOTE=YES is specified for CKD volumes (OPENDVCS=NO is specified or allowed to default), DEVN identifies the PPRC primary device paired with the PPRC secondary acting as FlashCopy source.

When ACTION=THAW is specified, DEVN must identify either a CKD access device for the open device to be thawed, or the CKD volume within the LSS to be thawed.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-byte field.

,SRCSERIAL=*srcserial*

When OPENDVCS(YES) or REMOTE(YES) is specified, SRCSERIAL identifies the storage control serial number that can include up to 10 digits, depending on the type of storage control.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 10-character field.

,SRCLSS=*srclss*

When OPENDVCS=YES or REMOTE=YES is specified, this identifies the two-digit hexadecimal value for the logical subsystem (LSS) for the device (ESS only).

Note: The LSS number is required if the storage control supports logical subsystems (like the ESS) and not allowed if the storage control does not support logical subsystems.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 1-character field.

,SRCDVC=*srcdvc*

When OPENDVCS=YES or REMOTE=YES is specified, this identifies the two-digit hexadecimal value for the device (either logical unit number or channel connection address).

To code: Specify the RS-type address, or address in register (2)-(12), of a required 1-character field.

,TGTSERIAL=*tgtserial*

When OPENDVCS=YES or REMOTE=YES is specified, TGTSERIAL identifies the storage control serial number that can include up to 10 digits, depending on the type of storage control.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 10-character field.

,TGTLSS=*tgtlss*
When OPENDVCS=YES or REMOTE=YES is specified, this identifies the two-digit hexadecimal value for the logical subsystem (LSS) for the device (ESS only).

Note: The LSS number is required if the storage control supports logical subsystems (like the ESS) and not allowed if the storage control does not support logical subsystems.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 1-character field.

,TGTDVC=*tgtdvc*
When OPENDVCS=YES or REMOTE=YES is specified, this identifies the two-digit hexadecimal value for the device (either logical unit number or channel connection address).

To code: Specify the RS-type address, or address in register (2)-(12), of a required 1-character field.

,TGTUCB=*tgtucb*
,TGTUCB=YES
is the name (RS-type) or address in register (2)-(12) of a 3-character field that contains YES or NO, indicating whether an MVS device number will be used for target addressability. Yes is the default.

YES
Indicates that the SDEVN and TDEVN keywords are being used to identify the source and target devices in the relationship, using MVS device numbers.

NO Indicates that an MVS device number will not be used for the target device (TDEVN keyword will not be used). Instead, the SDEVN, TGTSERIAL, TGTLSS, and TGTDVC keywords will be used.

Note: TGTUCB(NO) is not supported when OPENDVCS(YES) or REMOTE(YES) is specified.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 3-character field.

,SRCSSID=*srcssid*
When REMOTE=YES is specified without OPENDVCS=YES, this is the two-byte binary unsigned SSID of the subsystem where the FlashCopy request is to occur. This must be the same value as that specified for SSID on the PPRC secondary in the PPRC establish pair command.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional half-word field.

,OPENDVCS=*opendvcs*
,OPENDVCS=NO
Specifies whether Open System (fixed block) devices or CKD devices are addressed by this command. The values are:

NO The required parameters (SDEVN and TDEVN) identify CKD devices. The default is NO.

YES
Specifies that the required parameters (DEVN, SRCSERIAL, SRCLSS, SRCSSID, SRCDVC, TGTSERIAL, TGTLSS, TGTDVC) identify fixed block devices.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 3-character field.

,REMOTE=*remote*
,REMOTE=NO
Specifies whether the request is for a subsystem that is not directly attached to the issuing processor. The values are:

NO The request is directed to a device on a subsystem locally attached to the issuing processor. The default is NO.

YES
The request is directed to a remote subsystem using Inband subsystem functions. When REMOTE(YES) is specified, the request must be for a full volume.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 3-character field.

ACTION=*action*
This is the name (RS-type) or address in register (2)-(12) of an optional 12-byte (character) field that specifies an action to take. Valid values are COMMIT, REVERT, and THAW. There is no default. The value is left-justified and padded on the right with blanks. Valid values are:

COMMIT
This specifies that the last consistency group created by the Global Mirror session is committed to the target's current state, and reverting to the previous consistency group state is no longer possible. See "Moving a Global Mirror session to the recovery site in an unplanned outage" on page 446 for guidelines on the use of this option.

REVERT
Specifies that a rollback be performed to the state saved by a previous automatic FlashCopy establish command. The FlashCopy relationship is not removed (withdrawn), the FlashCopy target is rolled back to the previous consistency group created by the Global Mirror session. See "Moving a Global Mirror session to the recovery site in an unplanned outage" on page 446 for guidelines on the use of this option.

THAW
Specifies that a FlashCopy consistency group has been formed and application I/O can now take place on all volumes in the LSS where the previous FlashCopy Establish request was issued with the ACTION parameter specified as FREEZE. The DEVN parameter must be specified when ACTION=THAW is specified. SDEVN and TDEVN cannot be used.

If ACTION is specified, any extent specifications are ignored and DDSW is ignored

,RETINFO=*retinfo*
Is a required 100-character output field that returns detailed information about the results of **executing** the request. The name of the output field is a name (RS-type) or an address in register (2)-(12). The program returns

information about the results of scheduling the request in the RETCODE and RSNCODE fields. Align the area on a word boundary. The first 4-bytes contain the return code. The second 4-bytes contain the reason code. The remainder of the information depends on the return and reason codes. See the coding example about how to coordinate the RETCODE, RSNCODE, and RETINFO fields.

,DDSW=*ddsw*
,DDSW=NO

Is the name (RS-type) (or address in register (2)-(12) ASM only) of an optional 3-character input field that indicates if the deleted dataspace withdraw function (DDSW) is requested. The DDSW function causes two actions. First, within the specified source extents, all FlashCopy target tracks are immediately withdrawn from their FlashCopy relationship and second, all FlashCopy source tracks in a NOCOPY relationship are changed to a COPY (background copy) relationship. DDSW(YES) requires the source device, SDEVN, to be specified. If DDSW(YES) is specified, the target device parameters, TDEVN and TGTEXTENTS, are ignored.

NO indicates that the DDSW parameter is ignored.

YES indicates that the DDSW function is performed.

The default is NO.

,SRCEXTENTS=*srcextents* **| 0**

Is the name (RS-type) (or address in register (2)-(12) ASM only) of an optional variable character input field aligned on a fullword and containing the extent data structure described below. The extents in the extent data structure describe the source tracks on the source volume that participates in the FlashCopy withdraw process. If the SRCEXTENTS parameter is omitted and the source device SDEVN parameter is specified, all source tracks on the source volume participate in the FlashCopy withdraw process.

The following structure describes the extent data:

- A 4-byte field with a hexadecimal count of the source extent fields in the following list. This count field must have a value of 1 to 110 for devices in an ESS subsystem that supports volume FlashCopy. This count field must have a value of 1 to the maximum allowable extents for a device in an ESS subsystem that supports FlashCopy.

  This field may also contain a 4-character value of ALL. If this value is present, all tracks of the source device participates in the FlashCopy withdraw process.
- A 4-byte reserved field.
- A list of source extent fields. Each 8-byte extent field has two extents, the beginning source extent and the ending source extent on the source volume.

  The extent format is CCHH where CC is an unsigned 16-bit binary cylinder number. HH is an unsigned 16-bit binary track number.

  The extents in each source extent field contain the following:
  - A 4-byte beginning source extent.
  - A 4-byte ending source extent.

The default is 0.

, SPACEREL=*spacerel* **| NO**

This parameter is the name (RS-type) or address in register (2)-(12) of a three-character field that specifies whether the space-efficient volumes are released. The specified character value is left-justified and padded on the right with blanks.

NO Indicates that the space is not released if the volume is space efficient. This is the default.

YES
Indicates that, if the target volume is a space-efficient volume, when the withdraw is complete, any physical space for the space efficient volume is released.

Note:

1. If YES is specified and the target is not a space-efficient volume, the SPACEREL keyword is ignored. The results of the command cause the space-efficient volume become uninitialized and therefore unusable by the system until the volume is re-initialized.
2. If the request with SPACEREL(YES) is issued to a simplex volume that is a space-efficient volume, any physical space for the volume is released.
3. The allocated space is released asynchronously so may not be reported immediately.

,TGTEXTENTS=*tgtextents* **|** **0**
is the name (RS-type) (or address in register (2)-(12) ASM only) of an optional variable character input field aligned on a fullword and containing the extent data structure described below. The extents in the extent data structure describe the target tracks on the target volume that participates in the FlashCopy withdraw process. If the TGTEXTENTS parameter is omitted and the target device TDEVN parameter is specified, all target tracks on the target volume participate in the FlashCopy withdraw process.

The following structure describes the extent data:

- A 4-byte field with a hexadecimal count of the target extent fields in the following list. This count field must have a value of 1 to 110 for devices in an ESS subsystem that supports volume FlashCopy. This count field must have a value of 1 to the maximum allowable extents for a device in an ESS subsystem that supports FlashCopy.

  This field may also contain a 4-character value of ALL. If this value is present, all tracks of the target device participate in the FlashCopy withdraw process. This value is used only for FlashCopy withdraw target requests.
- A 4-byte reserved field.
- A list of target extent fields. Each 8-byte extent field has two extents, the beginning extent and the ending extent on the target volume that is part of this FlashCopy relationship.

  The extent format is CCHH where CC is an unsigned 16-bit binary cylinder number. HH is an unsigned 16-bit binary track number.

  The extents in each target extent field contain the following:
  - A 4-byte beginning target extent.
  - A 4-byte ending target extent.

The default is 0.

,ALET=*alet* **| 0**

Is the name (RS-type) (or address in register (2)-(12) ASM only) of an optional fullword input field that specifies the ALET value to use if the SRCBITMAP and TGTBITMAP fields have an ALET value different from the one associated with the caller.

The default is 0.

,ECB=*ecb*
,ECB=NO_ECB

Is the name (RS-type) (or address in register (2)-(12) ASM only) of an optional fullword input field that SDM posts for an asynchronous request. This field is ignored for synchronous requests.

The default is NO_ECB.

,WAITTIME=*waittime* **| 0**

Is the name (RS-type) (or address in register (2)-(12) ASM only) of an optional halfword input field containing how long in seconds SDM waits for a request to complete. A value of zero means do not time the request. If the time expires before the request completes, a value of 7039 (RQST_WAITTIME_EXPIRED - see ANTRQSTL) is returned. If the request is a synchronous request, the value is placed in the return code part of RETINFO. If the request is an asynchronous request, the value is used as the ECB post code.

The default is 0.

Note: When a FlashCopy NOCOPY relationship is ended, the track data on the target device is unpredictable and should not be used. If updates occur to source device tracks in the FlashCopy NOCOPY relationship, a copy of the source tracks from the point-in-time of the FlashCopy establish may or may not be written to the target device.

Tracks may be copied from the source to the target volume even if the source track is not changed. This includes the track that contains the volume label. Therefore, to avoid duplicate volume serial problems when the target device is later varied online, IBM recommends that you relabel the target volume after withdrawing a volume-level FlashCopy NOCOPY relationship.

,SUBCHSET=*subchset*
,SUBCHSET=0

Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is currently supported by the host system's processor and configured for the device.

If you specify a value of X'00' or omit this keyword, the command is issued to the device that is currently logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

To code: Specify the RS-type name, or address in register (2)-(12), of an optional 1-byte (character) field.

,TSUBCHSET=*tsubchset*
,TSUBCHSET=0

Used with the TDEVN parameter. *tsubchset* is the name (RS-type) or address in register (2)-(12) of a 1-character field that specifies the subchannel set is to be used to get information about the target device specified with the TDEVN parameter. This is the subchannel set for the device as defined in the Hardware Configuration Dialog (HCD). The values are determined by what is currently supported by the host system's processor and configured for the device.

If you specify a value of X'00' or omit this keyword, the command is issued to the device that is currently logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

Subparameters for REQUEST=QFRVOLS

REQUEST=QFRVOLS

Asks for the query Fast Replication volumes function to be executed.

,CTLVOL=*ctlvol*
,CTLVOL=NO_CTLVOL

Is the name (RS-type) (or address in register (2)-(12) ASM only) of an optional 6-character input field containing the 6-character volume serial of a control volume. The volume serial is left-justified and padded on the right with blanks. This parameter is required if the CTLDVC parameter is omitted or set to binary zero. This volume is used to determine the subsystem and device status and characteristics that are compared to the devices in the VOLLIST parameter list. This parameter must identify a device that is online to the host system where this QFRVOLS request is issued.

The default is NO_CTLVOL.

,CTLDVC=*ctldvc*
,CTLDVC=NO_CTLDVC

Is the name (RS-type) (or address in register (2)-(12) ASM only) of an optional halfword input field containing the 2 hexadecimal byte device number of a control volume. This parameter is required if the CTLVOL parameter is omitted or set to binary zero. This volume is used to determine the subsystem and device status and characteristics that are compared to the devices in the VOLLIST parameter list. This parameter must identify a device that is defined to the host system where this QFRVOLS request is being issued.

The default is NO_CTLDVC.

,VOLLIST=*vollist*

Is the name (RS-type) (or address in register (2)-(12) ASM only) of a required variable character input field containing the list of volumes that are examined for capability and eligibility in Fast Replication operations. Fast Replication operations include ESS FlashCopy and RVA Snapshot.

The size and format of this volume list area is described in the ANTQFRVL macro. For a description of the ANTQFRVL macro, see “ANTQFRVL macro” on page 737.

,VOLSRCTGT=*volsrctgt*

Is the name (RS-type) (or address in register (2)-(12) ASM only) of a required 6-character input field containing the 6-character identifier of which type of Fast Replication capability or eligibility is determined for each of the volumes in the VOLLIST. The identifier can be either one of the following, but cannot be omitted.

SOURCE indicates the volumes in the VOLLIST are checked for Fast Replication source volume capability or eligibility.

TARGET indicates the volumes in the VOLLIST are checked for Fast Replication target volume capability or eligibility.

There is no default.

,TGTPPRIM=*tgtpprim*
,TGTPPRIM=NO
Specifies whether the target in the FlashCopy relationship can be the primary in a PPRC pair. The values are:

NO The target in this FlashCopy relationship cannot be the primary in a PPRC pair. The default is NO.

YES
The target in this FlashCopy relationship can be the primary in a PPRC pair. This request proceeds normally to the specified target but the hardware ignores the PPRC status of the target.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 3-character field.

,RETINFO=*retinfo*
Is the name (RS-type) (or address in register (2)-(12) ASM only) of a required 100-character output field used to return detailed information about the results of executing the request. Information about the results of scheduling the request are returned in the RETCODE and RSNCODE fields. The area should be aligned on a word boundary. The first 4 bytes contain the return code and the second 4 bytes contain the reason code. The remainder of the information is dependent on the return and reason codes. See the coding example about how to coordinate the RETCODE, RSNCODE, and RETINFO fields.

,ALET=*alet* **| 0**
Is the name (RS-type) (or address in register (2)-(12) ASM only) of an optional fullword input field that specifies the ALET value be used for the QRYINFO field, the SRCEXTENTS field, the TGTEXTENTS field, the SRCBITMAP field, and the TGTBITMAP field. If the ALET parameter is specified, all of the fields are referenced using the ALET value. Otherwise, all of the output fields are referenced in the caller address space.

The default is 0.

SEFLC=*seflc*
,SEFLC=NO
This parameter is the name (RS-type) or address in register (2)-(12) of a three-character field that specifies whether the space-efficient volumes are included as eligible target volumes for a FlashCopy request. The specified character value is left-justified and padded on the right with blanks.

NO Indicates that space-efficient volumes are not eligible to be FlashCopy targets if VOLSRCTGT(TARGET) is specified. This is the default.

YES
Indicates that space-efficient volumes are eligible to be FlashCopy targets if VOLSRTGT(TARGET) is specified.

,ECB=*ecb*
,ECB=NO_ECB
Is the name (RS-type) (or address in register (2)-(12) ASM only) of an

optional fullword input field that SDM posts for an asynchronous request. This field is ignored for synchronous requests.

The default is NO_ECB.

,WAITTIME=*waittime* **| 0**

Is the name (RS-type) (or address in register (2)-(12) ASM only) of an optional halfword input field containing how long, in seconds, SDM waits for a request to complete. A value of zero means do not time the request. If the time expires before the request is complete, a value of 7039 (RQST_WAITTIME_EXPIRED - see ANTRQSTL) is returned. If the request is a synchronous request, the value is placed in the return code part of RETINFO. If the request is an asynchronous request, the value is used as the ECB post code.

The default is 0.

,PRESMIR=*presmir*
,PRESMIR=NO

Is an optional 4-character input field that contains a keyword that specifies whether Preserve Mirror configuration checking is to be done as part of verifying the eligibility of the target. The input field is an RS-type name or an address in register (2)-(12). The value is left-justified and padded on the right with blanks.

NO Indicates that Preserve Mirror checking is not needed.

REQ
Indicates that Preserve Mirror checking must be done. If the device is not Preserve Mirror-eligible, the device is not eligible.

PREF
Indicates that Preserve Mirror checking must be done. Target volumes are eligible even if a Preserve Mirror operation cannot be done, as long as the device is FlashCopy-eligible.

The default is NO.

Subparameters for REQUEST=QHA

REQUEST=QHA

Requests the QHA function, to query a device and determine where path groups are established. The device is considered online where path groups are established.

,DEVN=*devno*

Is a required input parameter field containing the MVS device number to which the command is to be issued.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-byte field.

,QRYSIZE=*qrysize*

Is a required input parameter field containing the length of the QRYINFO area.

To code: Specify the RS-type address, or address in register (2)-(12), of a halfword input field.

,ALET=*alet* **|** **0**

Is an optional parameter input field that specifies the ALET value to use if the QRYINFO field has an ALET value different from the one associated with the caller.

The default is 0.

To code: Specify the RS-type name, or address in register (2)-(12), of a fullword field.

,CCA=*cca*

Is an optional input parameter field that specifies the 2-digit hexadecimal value of the CCA for the device to be queried. The default is 00 (no CCA). It is required if you specify TGTUCB=NO, and ignored otherwise.

To code: Specify the RS-type address, or address in register (2)-(12), of a 1-character field.

,ECB=*ecb*

,ECB=NO_ECB

Is an optional fullword input field that SDM posts for an asynchronous request. The name of the input field is a name (RS-type) or address in register (2)-(12). The program ignores this field for synchronous requests.

The default is NO_ECB.

,LSS=*lss*

Is an optional input parameter field that specifies the 2-digit hexadecimal value of the LSS that is to be queried. The default is 00 (no LSS). It is used only if you specify TGTUCB=NO.

To code: Specify the RS-type address, or address in register (2)-(12), of a 1-character field.

,QRYINFO=*qryinfo*

specifies the name of a variable-character output field that returns the result of the query.

If the field is not big enough to contain the complete result, the program places a return code of (RQST_QHA_QRYSIZE_TOO_SMALL) in the return code part of RETINFO and places the total space required in the reason code part of RETINFO.

If the field is big enough to contain the complete result, the program places a return code of (RQST_QHA_QRYSIZE_BIG_ENOUGH) in the return code part of RETINFO and places the total amount of space actually used in the reason code part of RETINFO.

You can use the optional ALET parameter to specify the location of the QRYINFO field.

If the channel command is not supported by the storage control where it was issued, the program places a return code of (RQST_QHA_NOT_SUPPORTED) in the return code part of RETINFO.

To code: Specify the RS-type name, or address in register (2)-(12), of a required variable-character output field.

,RETINFO=*retinfo*

Is a required 100-character output field that the program uses to return detailed information about the results of **executing** the request. The name of the output field is a name (RS-type) or address in register (2)-(12). The program returns information about the results of **scheduling** the request in the RETCODE and RSNCODE fields. Align the area on a word boundary. The first 4 bytes contain the return code. The second 4 bytes contain the reason code. The remainder of the information depends on the return and reason codes. See the coding example about how to coordinate the RETCODE, RSNCODE, and RETINFO fields.

,SUBCHSET=*subchset*
,SUBCHSET=0

Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is currently supported by the host system's processor and configured for the device.

If you specify a value of X'00' or omit this keyword, the command is issued to the device that is currently logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

To code: Specify the RS-type name, or address in register (2)-(12), of an optional 1-byte (character) field.

,TGTUCB=*tgtucb*
,TGTUCB=YES

specifies whether the query results are requested for a FlashCopy device with no target UCB.

YES
Indicates that the query results are requested for the device specified by DEVN. The LSS and CCA keywords are ignored. YES is the default.

NO Indicates that LSS and CCA keywords should be used to identify the device to be queried. You must also use the DEVN keyword to specify a device within the same SFI as the device to be queried.

To code: Specify the RS-type address, or address in register (2)-(12), of a 3-character field.

,WAITTIME=*waittime* **| 0**

Is an optional halfword input field that contains how long in seconds SDM waits for a request to complete. The name of the input field is a name (RS-type) or address in register (2)-(12). A value of zero specifies the requests should not be timed. If the time expires before the request completes, the program returns a value of 7039 (RQST_WAITTIME_EXPIRED). For information about RQST_WAITTIME_EXPIRED, see ANTRQSTL. If the request is a synchronous request, the program places the value in the return code part of RETINFO. If the request is an asynchronous request, the program uses the value as the ECB post code.

The default is 0.

Subparameters for REQUEST=RQUERY

REQUEST=RQUERY

Asks for the status on a Global Mirror for ESS session.

,SNBR=*sessno*

Is the name (RS-type) (or address in register (2)-(12) ASM only) of a required 1-byte value between 01 and FF identifying the session number.

You may specify 0 with the GMLSTAT, GMPSTAT, or DVCSTAT ACTION parameters to query all Global Mirror sessions defined to the LSS.

,VOLSER=*volno*
Is the name (RS-type) (or address in register (2)-(12) ASM only) of a required 6-character MVS volume serial number of the device to be used for I/O. For CKD, *volno* must be in the same LSS as the target of the query. For FB, *volno* must be in the same cluster as the target of the query.

,DEVN=*devn*
A required input parameter field containing the binary device number to use for I/O.

The allowed values specified in this keyword are a 4 hexadecimal character MVS device number. This keyword is mutually exclusive with the VOLSER keyword.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-byte field.

,ACTION=*action*
Is the name (RS-type) (or address in register (2)-(12) ASM only) of a required 12-character value identifying the specific action to be performed.

STAT4ALSS
This query provides performance counters, consistency group information, FlashCopy information, and out-of-sync tracks for the specified session within the specified LSS. This action will result in only unformatted output. The output is mapped by ANTRQMAP TYPE=STAT4A.

STAT4AESS
This query provides performance counters, consistency group information, FlashCopy information, and out-of-sync tracks for the specified session within the ESS the specified LSS is in. This action will result in only unformatted output. The output is mapped by ANTRQMAP TYPE=STAT4A.

STAT4ACGRP
This query provides performance counters, consistency group information, FlashCopy information, and out-of-sync tracks for the specified session within the ESS the specified LSS is in. This action will result in only unformatted output. The output is mapped by ANTRQMAP TYPE=STAT4A.

STAT4BLSS
This query provides out-of-sync information for the specified session within the specified LSS. This action will result in only unformatted output. The output is mapped by ANTRQMAP TYPE=STAT4B.

STAT4BESS
This query provides out-of-sync information for the specified session within the ESS the specified LSS is in. This action will result in only unformatted output. The output is mapped by ANTRQMAP TYPE=STAT4B.

STAT4C
This query provides summary information on all sessions in the ESS. This action results in only unformatted output. The output is mapped by ANTRQMAP TYPE=STAT4C.

STAT51
This query provides detailed performance information on all the sessions in an ESS. This action results in only unformatted output. The output is mapped by ANTRQMAP TYPE=STAT51. For more information, see the note that follows these values, at Note for ACTION=STAT51 and ACTION=GMLSTAT.

GMLSTAT
This query provides a formatted report, either to a data set or to the QRYINFO, with summary information pertaining to the Global Mirror session. For more information, see the note that follows these values, at Note for ACTION=STAT51 and ACTION=GMLSTAT.

GMPSTAT
This query provides a formatted report, either to a data set or to the QRYINFO, with summary information pertaining to the LSS specified in the request and the Global Mirror session.

DVCSTAT
This query provides a formatted report, either to a data set or to the QRYINFO, with information pertaining to all devices in the LSS joined to the Global Mirror session.

Note: For ACTION=STAT51 and ACTION=GMLSTAT, the control unit returns information for multiple sessions, with detail for sessions that were started in the same cluster as the device where the query was issued, and a status of NotOwnedCL, for sessions that were started in the other cluster, or NotMaster, for subordinate sessions. Information returned for sessions not owned by the cluster where the query was issued or for a subordinate session consists of:

- Session ID
- CG interval time
- Coordination interval time
- Max drain time
- Master CU sequence number

If the IBM storage management software specified the SYSPLEXNAME keyword on START or RESUME, the following are included:

- Current sysplex clock time
- Drift
- Last Good CG sysplex clock time
- Sysplex name.

When multiple sessions are started for an SFI, ACTION=GMLSTAT with SNBR of 00 returns information for multiple sessions, separated by a line of asterisks (*). Specifying a SNBR other than 00 displays only query information for that session. The default is 00 (show all sessions in the query).

,LSSNBR=*lss*
lss is the name (RS-type) (or address in register (2)-(12) ASM only) of an optional 1-byte field specifying the LSS number for which the report is to be generated by an RQUERY request. Valid values are 00-FF.

,DVCNBR=*dvc*
dvc is the name (RS-type) (or address in register (2)-(12) of an optional 1-byte field specifying the device for which the query is being issued. When LSSTYPE is FB, this parameter is required and is the device LUN. When optionally specified for an LSSTYPE of CKD, this value is the device CCA.

,LSSTYPE=*typ*
,LSSTYPE=CKD
typ is the name (RS-type) (or address in register (2)-(12) of an optional 3-character field specifying the type of LSS. Valid values are FB (left-justified and padded on the right with a blank) or CKD. This parameter is required for FB. CKD is the default.

,QRYSIZE=*size*
size is the name (RS-type) or address in register (2)-(12) of a required half word input field that contains the length of QRYINFO area. This is mutually exclusive with the DSNAME keyword.

,QRYINFO=*info*
info is the name (RS-type) or address in register (2)-(12) of a required variable character output field to be used to return the results of the query. This keyword is mutually exclusive with DSNAME keyword.

,DSNAME=*dsn*
dsn is the name (RS-type) or address in register (2)-(12) of a 44-character input field containing the name of a data set that the query output will be placed in. The data set must already be allocated with a fixed record format and a logical record length of at least 79 bytes. The value is-left justified and padded on the right with blanks. The keyword is mutually exclusive with the QRYSIZE and QRYINFO keywords.

,DSDISP=*disp*
,DSDISP=OLD
disp is the name (RS-type) or address in register (2)-(12) of an optional 3-character input field containing a parameter that tells the disposition of the data set specified in the DSNAME keyword. The value is left justified and padded on the right with blanks. A value of binary zeros indicates that the default is to be used. Valid values that can be specified are:

OLD
A disposition of old. This is the default.

MOD
A disposition of mod.

SHR
A disposition of shr.

,SUBCHSET=*subchset*
,SUBCHSET=0
Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is currently supported by the host system's processor and configured for the device.

If you specify a value of X'00' or omit this keyword, the command is issued to the device that is currently logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

To code: Specify the RS-type name, or address in register (2)-(12), of an optional 1-byte (character) field.

Subparameters for REQUEST=RSESSION

REQUEST=RSESSION
Asks for the Global Mirror for ESS function to be executed.

,SNBR=*sessno*
Is the name (RS-type) (or address in register (2)-(12) ASM only) of a required 1-byte value between 01 and FF, identifying the session number.

,VOLSER=*volno*
Is the name (RS-type) (or address in register (2)-(12) ASM only) of a required 6-character MVS volume serial number of the device to be used for I/O. For CKD, *volno* must be in the LSS targeted by the command. For FB, *volno* must be in the same cluster as the target of the command.

,DEVN=*devn*
A required input parameter field containing the binary device number to use for I/O.

The allowed values specified in this keyword are a 4 hexadecimal character MVS device number. This keyword is mutually exclusive with the VOLSER keyword.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-byte field.

,ACTION=*action*
Is the name (RS-type) (or address in register (2)-(12) ASM only) of a required 12-character value identifying the specific action to be performed.

DEFINE
Define a new Global Mirror session .

UNDEFINE
Remove the specified Global Mirror session.

START
Start forming consistency groups for the specified Global Mirror session.

RESUME
Restart the specified Global Mirror session with new tuning values.

If the Global Mirror session was paused in a consistent manner using CGPAUSE, RESUME automatically unsuspends and resyncs the Global Copy pairs in the session.

PAUSE
Pause the specified Global Mirror session.

STOP Terminate the specified Global Mirror session.

CGPAUSE
Pause the Global Mirror session and suspend the Global Copy pairs in the session so that the secondaries are consistent.

,LSSTYPE=*lsstype*
,LSSTYPE=CKD
Is the name (RS-type) (or address in register (2)-(12) ASM only) of an optional 3-byte (character) field identifying the type of devices in the LSS. The value is left-justified and padded on the right with blanks. This parameter defaults to

CKD and is only required when addressing open devices with ACTION DEFINE or UNDEFINE requests. Valid values are:

FB The LSS contains open, or fixed block devices.

CKD
The LSS contains count key data, or System z-attached devices.

,LSSNBR=*lss*
Is the name (RS-type) (or address in register (2)-(12) ASM only) of an optional 1-byte field identifying the LSS this command targets. This parameter is required for DEFINE and UNDEFINE actions, and ignored for the remaining actions.

,ESSSERIAL=*ess*
Is the name (RS-type) (or address in register (2)-(12) ASM only) of an optional 12-byte field that identifies the serial number of the ESS storage control that this request targets. This parameter is required for DEFINE and UNDEFINE actions, and ignored for the other actions.

,CGINTERVAL=*cgint*
,CGINTERVAL=0
Is the name (RS-type) (or address in register (2)-(12) ASM only) of an optional half-word field containing a hexadecimal number between X'0000' and X'FFFF' that specifies the consistency group interval time in seconds. If this field is set to zero, consistency groups will be formed continuously. This parameter is optional on a START action, required on a RESUME action, and ignored for all other actions. If this parameter is specified, then CGDRAIN and COORDINTERVAL are required.

,CGDRAIN=*drain*
,CGDRAIN=30
Is the name (RS-type) (or address in register (2)-(12) ASM only) of an optional half-word field containing a hexadecimal number between X'0000' and X'FFFF' that specifies the maximum consistency group drain time in seconds. If this field is set to zero, the default value of 30 seconds is used. This parameter is optional on a START action, required on a RESUME action, and ignored for all other actions. If this parameter is specified, then CGINTERVAL and COORDINTERVAL are required.

,COORDINTERVAL=*coord*
,COORDINTERVAL=50
Is the name (RS-type) (or address in register (2)-(12) ASM only) of an optional half-word field containing a hexadecimal number between X'0000' and X'FFFF' that specifies the consistency group maximum coordination interval in milliseconds. If this field is set to zero, default of 50 ms is used. This parameter is optional on a START action, required on a RESUME action, and ignored for all other actions. If this parameter is specified, then CGINTERVAL and CGDRAIN are required.

,MSSERIAL=*msser*
Is the name (RS-type) (or address in register (2)-(12) ASM only) of an optional 12-byte field that specifies the serial number of the master storage control. This parameter is required on a START, RESUME, PAUSE, or STOP actions and ignored for other actions.

,SBINFO=*nMssidSssidSerialnoMssidSssidSerialno...*
Is the name (RS-type) (or address in register (2)-(12) of an optional variable length field that specifies a list of ESS subordinate sets (topography) in the session. The values in a set come from the corresponding values that were

used for the PESTPATH command connecting the master storage control and storage control subordinates. This parameter is required for START, RESUME, PAUSE, and STOP actions and ignored for the other actions.

n Is the number of ESS subordinate sets, and is one byte long.

Each subordinate set includes:

Mssid
The two-byte master SSID where the control path originates connecting the subordinate storage control.

Sssid
The two-byte subordinate SSID where the control path from the master storage control is established.

Serialno
A 12-byte subordinate storage control serial number.

,MASTER=*master*
,MASTER=YES
Is the name (RS-type) (or address in register (2)-(12) ASM only) of an optional 3-character field specifying whether this command is being issued to a master storage control or to a subordinate storage control. Valid values are YES and NO. The value is left-justified and padded on the right with blanks. This parameter is optional for a STOP action and is ignored for all other actions. Pause is always done to the Master. Stop is issued to the master unless there is an error, in which case Stop may be issued to each subordinate. If you stop an individual subordinate while a session is running, it will cause the session to go into a fatal status. The session will have to be stopped, corrected, then started again.

YES
Indicates the action is issued to the master storage control. This is the default.

No Indicates the action is issued to a subordinate storage control.

,FORCE=*force*
,FORCE=NO
Is the name (RS-type) (or address in register (2)-(12) ASM only) of an optional 3-character field specifying what kind of cleanup is to be done. Valid values are YES and NO. The value is left-justified and padded on the right with blanks. This parameter is optional for a STOP action, ignored for all other actions.

YES
Indicates that the STOP is to be performed even if a consistent copy of data cannot be created at the recovery site.

No Indicates the stop will not complete until a consistent copy of the data has been formed. This is the default.

,SUBCHSET=*subchset*
,SUBCHSET=0
Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is currently supported by the host system's processor and configured for the device.

If you specify a value of X'00' or omit this keyword, the command is issued to the device that is currently logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

To code: Specify the RS-type name, or address in register (2)-(12), of an optional 1-byte (character) field.

Subparameters for REQUEST=RVOLUME

REQUEST=RVOLUME
Specifies volumes to be shadowed in the Global Mirror for ESS session.

,SNBR=*sessno*
Is the name (RS-type) (or address in register (2)-(12) ASM only) of a required 1-byte value between 00 and FF, identifying the session number.

,VOLSER=*volno*
Is the name (RS-type) (or address in register (2)-(12) ASM only) of a required 6-character MVS volume serial number of the device to be used for I/O. For CKD, *volno* must be in the LSS as the volumes targeted by the command. For FB, *volno* must be in the same cluster as the volumes targeted by the command.

,DEVN=*devn*
A required input parameter field containing the binary device number to use for I/O.

The allowed values specified in this keyword are a 4 hexadecimal character MVS device number. This keyword is mutually exclusive with the VOLSER keyword.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-byte field.

,LSSTYPE=*lsstype*
,LSSTYPE=CKD
Is the name (RS-type) (or address in register (2)-(12) ASM only) of an optional 3-byte (character) field identifying the type of devices in the LSS. The value is left-justified and padded on the right with blanks. This parameter defaults to CKD and is only required when addressing open devices with ACTION DEFINE or UNDEFINE requests. Valid values are:

FB The LSS contains open, or fixed block devices.

CKD
The LSS contains count key data, or System z-attached devices.

,LSSNBR=*lss*
Is the name (RS-type) (or address in register (2)-(12) ASM only) of an optional 1-byte field identifying the LSS this command targets. This parameter is required for DEFINE and UNDEFINE actions, and ignored for the remaining actions.

,ESSSERIAL=*ess*
Is the name (RS-type) (or address in register (2)-(12) ASM only) of an optional 12-byte field that identifies the serial number of the ESS storage control that this request targets. This parameter is required for DEFINE and UNDEFINE actions, and ignored for the other actions.

,ACTION=*action*

Is the name (RS-type) (or address in register (2)-(12) ASM only) of a required 12-character value identifying the specific action to be performed.

JOIN
Indicates that volume sets specified in the VOLLIST or VOLRANGE parameter are to be added to the Global Mirror session specified by SNBR.

REMOVE
Indicates that volume sets specified in the VOLLIST or VOLRANGE parameter are to be removed from the Global Mirror session specified by SNBR.

,VOLLIST=*vlist*

Is the name (RS-type) (or address in register (2)-(12) ASM only) of a variable length field that identifies a list of Channel Connection Addresses (CCAs) or Logical Unit Numbers (LUNs) that are involved in the action. The first byte, *n*, indicates how many CCAs or LUNs are specified. The remaining bytes contain one-byte addresses. Up to 256 CCAs or LUNs can be specified. All CCAs/LUNs must be defined in the LSS specified in the LSSNBR parameter. This parameter is mutually exclusive with VOLRANGE , MTVOLLIST and MTVOLRANGE.

,VOLRANGE=*vrange*

Is the name (RS-type) (or address in register (2)-(12) ASM only) of a variable length field that identifies a list of ranges of Channel Connection Addresses (CCAs) or Logical Unit Numbers (LUNs) that are involved in the action. The first byte, *n*, indicates how many ranges are specified. The remaining bytes contain two-byte pairs. The first byte is the CCA or LUN of the first device in the range and the second byte is the CCA or LUN of the last device in the range. Up to 256 ranges of CCAs or LUNs can be specified. All CCAs/LUNs must be defined in the LSS specified in the LSSNBR parameter. This parameter is mutually exclusive with VOLLIST, MTVOLLIST and MTVOLRANGE.

,MTVOLLIST=*vlist*

Is the name (RS-type) (or address in register (2)-(12) ASM only) of a variable length field that identifies a list of CCAs or LUNs, and their associated PPRC secondary information (serial number, LSS, and CCA/LUN), that are to be joined to or removed from the Global Mirror session.

The first byte, *n*, indicates how many primary CCAs or LUNs are specified. The remaining bytes are treated as an array with *n* entries. Each array entry consists of the following:

pcca
Specifies the one-byte primary CCA (or LUN)

secser#
Specifies the 12-character serial number of the secondary device's SFI for the Global Copy pair

slss
Specifies the one-byte LSS number of the secondary device for the Global Copy pair

scca
Specifies the one-byte CCA (or LUN) number of the secondary device for the Global Copy pair

Specify up to 255 device sets. This parameter is mutually exclusive with MTVOLRANGE, VOLLIST and VOLRANGE. When there are multiple secondaries, you must use MTVOLLIST or MTVOLRANGE.

,MTVOLRANGE=*vrange*

Is the name (RS-type) (or address in register (2)-(12) ASM only) of a variable length field that identifies a list of primary device ranges in an LSS and their associated PPRC secondary information (serial number, LSS, and CCA/LUN) that are to be added to or removed from the Global Mirror session. The first byte, *n*, indicates how many primary CCAs or LUNs are specified. The remaining bytes are treated as an array with *n* entries. Each array entry consists of the following:

pcca1 Specifies the first primary CCA (or LUN) in a range of contiguous primary devices

pccax Specifies the last primary CCA (or LUN) in a range of contiguous primary devices

secser# Specifies the serial number of the secondary device's SFI for the Global Copy pair

slss Specifies the LSS number of the secondary device for the Global Copy pair

scca1 Specifies the first CCA (or LUN) of the secondary device for the Global Copy pair. The last secondary CCA is derived from the number of devices specified in the *pcca1*-*pccax* range.

Specify up to 255 device set ranges (*n*=FF). This parameter is mutually exclusive with VOLLIST, VOLRANGE and MTVOLLIST. When there are multiple secondaries, you must use MTVOLLIST or MTVOLRANGE.

,SUBCHSET=*subchset*
,SUBCHSET=0

Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is currently supported by the host system's processor and configured for the device.

If you specify a value of X'00' or omit this keyword, the command is issued to the device that is currently logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

To code: Specify the RS-type name, or address in register (2)-(12), of an optional 1-byte (character) field.

Subparameters for REQUEST=STATESAVE

REQUEST=STATESAVE

Takes a state save (standard or non-disruptive) for the storage facility image (SFI) in which a specified device resides.

One statesave is allowed every 5 minutes. However, only 10 non-disruptive statesaves are allowed in a 24-hour period.

,DEVN=*devno*

Is a required input parameter field containing the 2-byte (4 hexadecimal digits) number of the device to which the STATESAVE command is to be issued.

Because a state save is applicable to an entire SFI, the device to which the command is issued does not have to be the device on which an error occurred. IBM recommends that an alternate device in the same SFI be used as the target for the state save, as the device in error may not be available.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-byte field.

,NDSS=*statesave*
,NDSS=YES
Is an optional input parameter field that indicates whether a non-disruptive state save should be taken.

YES
Requests a non-disruptive state save. This does not cause a warm start of the SFI. This is the default.

NO Requests a standard state save, with a resulting warm start in the SFI.

To code: Specify the RS-type address, or address in register (2)-(12), of a 3-character field.

,SUBCHSET=*subchset*
,SUBCHSET=0
Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is currently supported by the host system's processor and configured for the device.

If you specify a value of X'00' or omit this keyword, the command is issued to the device that is currently logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

To code: Specify the RS-type name, or address in register (2)-(12), of an optional 1-byte (character) field.

Use the following optional input parameter fields to describe the state save. The values are not checked or validated.

,CALLER=*caller*
Is an optional input parameter field that specifies the name of the caller requesting the state save. The default is INTERNAL (the internal caller).

To code: Specify the RS-type address, or address in register (2)-(12), of a 8-byte field.

,CCA=*cca*
Is an optional input parameter field that specifies the CCA associated with the device where the initial error occurred. The default is 00 (no CCA).

To code: Specify the RS-type address, or address in register (2)-(12), of a 1-byte field.

,DIAGREAS=*diagreason*
Is an optional input parameter field that specifies a diagnostic reason code to be associated with the state save. This can be used to identify the problem that prompted the state save. The default is 00.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-byte field.

,DIAGRETC=*diagretc*
Is an optional input parameter field that specifies a diagnostic return code to be associated with the state save. This can be used to identify the problem that prompted the state save. The default is 00.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-byte field.

,FUNC=*function*
Is an optional input parameter field that identifies the function being performed at the time the state save was requested. This is a free-form field that is not checked or validated. You can specify anything that may help explain the conditions at the time that the state save was requested. The default is no function.

To code: Specify the RS-type address, or address in register (2)-(12), of a 20-character field.

,LSS=*lss*
Is an optional input parameter field that specifies the LSS associated with the device where the initial error occurred. The default is 00 (no LSS).

To code: Specify the RS-type address, or address in register (2)-(12), of a 1-byte field.

,SEQNO=*seqno*
Is an optional input parameter field that specifies the XRC read record set sequence number associated with the initial error. The default is 00 (no sequence number).

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-byte field.

,SESSION=*session*
Is an optional input parameter field that specifies the XRC session associated with the state save. The default is 00 (no session).

To code: Specify the RS-type address, or address in register (2)-(12), of a 1-byte field.

,TIME=*timestamp*
Is an optional input parameter field that specifies the time, in STCK format, associated with the state save. The default is the time at which the state save I/O request is issued.

To code: Specify the RS-type address, or address in register (2)-(12), of an 8-byte field.

,TITLE=*title*
Is an optional input parameter field that specifies a title for the state save. This is a free-form text field that is not checked or validated. You can specify anything that helps explain the conditions at the time that the state save was requested. The default is no title text.

To code: Specify the RS-type address, or address in register (2)-(12), of a 50-character field.

,TYPE=*functiontype*
Is an optional input parameter field that specifies the type of function that was in control at the time the state save was requested. The default is 00 (no type). The values for *functiontype* are:

CC Concurrent Copy

FC FlashCopy

FREEZE
Freeze

GC Global Copy

GM Global Mirror

MM Metro Mirror

PATHS
Establish or delete paths

RUN
Run

SS SnapShot

XRC
Extended Remote Copy

XRCER
XRC Enhanced Reader

To code: Specify the RS-type address, or address in register (2)-(12), of a 6-character field.

Parameter descriptions for ILK=PPRC

When you invoke the ANTRQST macro for peer-to-peer remote copy, you can use the following REQUEST parameters. For a list of the Common End Codes associated with each request within this ILK, see "Common end codes for REQUESTS within each ILK" on page 735.

Subparameters for REQUEST=FENCE

REQUEST=FENCE

| Asks for the FENCE function of PPRC to be executed. Use this function to set or reset a Soft fence or a SPID fence.

Soft Fence can be used to prevent unintended access of a device. To clear the soft fenced state for a device, you can use the ICKDSF CONTROL command with the SCOPE parameter. For more information, refer to CONTROL in *Device Support Facilities (ICKDSF) User's Guide and Reference*.

| SPID Fence can be used to prevent SPID commands from grouping the device and to prevent any hosts from bringing the device online until the SPID fence has been reset.

| The request can be applied to a device, a set of devices in a Logical Subsystem (LSS), or all devices in an LSS. The current Fence status of a device can be seen in the results of both the QHA query and PQSCSTAT query.

After recovery, a Failback performed by storage management software or by a user command automatically unfences the secondary volume.

,DEVN=*devn*
A required input parameter field containing the device number to which the FENCE command is issued.

If you also specify OPENDVCS=YES then *devn* must identify a CKD device defined to the host system making this request.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-byte field.

,ACTION=*action*

A required input parameter field that specifies an action to take.

FENCE
: Specifies that a Soft Fence operation is requested.

UNFENCE
: Specifies that a Reset Soft Fence operation is requested.

SPIDFENCE
: Specifies that a SPID Fence operation is requested.

SPIDUNFENCE
: Specifies that a Reset SPID Fence operation is requested.

To code: Specify the RS-type address, or address in register (2)-(12), of a 4-byte field.

,DVC=*dvc*

An optional input parameter field that indicates the binary logical unit number (LUN) for the device to be fenced or unfenced. This keyword is required if you specify OPENDVCS=YES and SCOPE=DEV; it is ignored otherwise.

To code: Specify the RS-type address, or address in register (2)-(12), of a 1-byte field.

,ECB=*ecb*
,ECB=NO_ECB

An optional input parameter field that SDM posts for an asynchronous request. This field is ignored for synchronous requests. The default is NO_ECB.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,LSS=*lss*

An optional input parameter field that indicates the binary logical subsystem number for the FB device to be fenced or unfenced. This keyword is required if you specify OPENDVCS=YES; it is ignored otherwise.

To code: Specify the RS-type address, or address in register (2)-(12), of a 1-byte field.

,MASK=*mask*

An optional input parameter field that specifies a 32-byte mask, each bit of which represents a device on which to perform a soft fence operation. Bit 0 of the first byte represents device 0. Bit 7 of the last byte represents device 255. MASK is required when you specify SCOPE=MASK. Otherwise, it is ignored.

To code: Specify the RS-type address, or address in register (2)-(12), of a 32-byte field.

,OPENDVCS=*opendvcs*
,OPENDVCS=NO

An optional input parameter field containing a keyword that indicates if the request is for Open System (Fixed Block) devices. The value is left-justified and padded on the right with blanks. If the field contains binary zeros, the OPENDVCS keyword is treated as omitted.

YES
Specifies that the request applies to an Open System LSS or device.

NO Specifies that the device specified with DEVN is a CKD device. This is the default.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 3-character field.

,RETINFO=*retinfo*
An optional output parameter field that is used to return detailed information about the results of *executing* the request. Information about the results of *scheduling* the request are returned in the RETCODE and RSNCODE fields. Align the area on a word boundary. The first 4 bytes contain the return code and the second 4 bytes contain the reason code. The remainder of the information is dependent on the return and reason codes. See "SDM API usability guide" on page 575 about how to coordinate the RETCODE, RSNCODE, and RETINFO fields.

To code: Specify the RS-type address, or address in register (2)-(12), of a 100-character field.

,SCOPE=*scope*
,SCOPE=DEV
An optional input parameter field that specifies the scope of the fence operation. The values and the scopes that they specify are:

DEV
The specified device only. This is the default.

LSS
All devices in the LSS of the device specified in the DEVN keyword or the LSS keyword if OPENDVCS=YES.

MASK
All devices specified by with MASK. You must also specify the MASK keyword with a non-zero mask.

To code: Specify the RS-type address, or address in register (2)-(12), of a 4-character field.

,SUBCHSET=*subchset*
,SUBCHSET=0
Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is currently supported by the host system's processor and configured for the device.

If you specify a value of X'00' or omit this keyword, the command is issued to the device that is currently logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

To code: Specify the RS-type name, or address in register (2)-(12), of an optional 1-byte (character) field.

| **,TOKEN=***token*
| An 8-byte field containing the host token. The token is a unique identifier

used to identify the owner of the SPID fence for the device. The TOKEN keyword is valid only for ACTION(SPIDFENCE) and ACTION(SPIDUNFENCE). The TOKEN is ignored for all other ACTION keyword values.

,WAITTIME=*waittime* **| 0**

An optional input parameter field containing how long in seconds SDM waits for a request to complete. A value of zero indicates that the system does not time the request. If time expires before the request completes, a value of 7039 (RQST_WAITTIME_EXPIRED) is returned. For information about RQST_WAITTIME_EXPIRED, see ANTRQSTL. If the request is a synchronous request, the value is placed in the return code part of RETINFO. If the request is an asynchronous request, the value is used as the ECB post code.

Subparameters for REQUEST=LEVEL

REQUEST=LEVEL

Asks for the level of ANTRQST that is installed on the system.

,RETINFO=*retinfo*

is the name (RS-type) (or address in register (2)-(12) ASM only) of a required 100-character output field used to return the level of ANTRQST installed on the system. If ANTRQST is not installed, return code 7000 (RQST_PC_NUMBER_ZERO - see ANTRQSTL) is returned in register 15 and placed in the RETCODE area, if specified. If ANTRQST is installed, a level value is placed in the first 4-bytes of the RETINFO field. This level value indicates which ANTRQST functions, keywords and parameters are supported. For more information about the LEVEL request output, see the prolog comments in the ANTRQSTL macro (shipped with the ANTRQST macro).

Subparameters for REQUEST=PDELPAIR

REQUEST=PDELPAIR

Asks for the CDELPAIR function of PPRC to be executed.

,DEVN=*devn*

A required input parameter field containing the binary device number to use for I/O.

If OPENDVCS=YES is specified, then *devn* identifies a System z device to act as a CKD access volume. This device must be located in the same subsystem cluster as the primary device.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-byte field.

,PCCA=*pcca*

A required input parameter field containing the binary channel connection address of the primary CKD device. For open (fixed block) devices, this specifies the logical unit number (lun).

To code: Specify the RS-type address, or address in register (2)-(12), of a 1-byte field.

,PSERIAL=*pserial*

A required input parameter field containing the EBCDIC serial number of the primary storage control.

To code: Specify the RS-type address, or address in register (2)-(12), of a 12-character field.

,PSSID=*pssid*
A required input parameter field containing the binary subsystem ID of the primary storage control. When OPENDVCS=YES this value should be specified as X'FFFF'.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-byte field.

,SCCA=*scca*
A required input parameter field containing the binary channel connection address of the secondary device. For open (fixed block) devices, this specifies the logical unit number (lun).

To code: Specify the RS-type address, or address in register (2)-(12), of a 1-byte field.

,SSERIAL=*sserial*
A required input parameter field containing the EBCDIC serial number of the secondary storage control.

To code: Specify the RS-type address, or address in register (2)-(12), of a 12-character field.

,SSSID=*sssid*
A required input parameter field containing the binary subsystem ID of the secondary storage control. When OPENDVCS=YES this value should be specified as X'FFFF'.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-byte field.

,RETINFO=*retinfo*
A required output parameter field that is used to return detailed information about the results of *executing* the request. Information about the results of *scheduling* the request are returned in the RETCODE and RSNCODE fields. Align the area on a word boundary. The first 4-bytes contain the return code and the second 4-bytes contain the reason code. The remainder of the information is dependent on the return and reason codes. See “SDM API usability guide” on page 575 about how to coordinate the RETCODE, RSNCODE, and RETINFO fields.

To code: Specify the RS-type address, or address in register (2)-(12), of a 100-character field.

,PLSS=*plss*
,PLSS=NO_PLSS
An optional input parameter field containing the binary logical subsystem number within the primary storage subsystem. This keyword is required for storage subsystems with logical subsystems and is invalid for 3990 and other 3990–compatible storage controls.

NO_PLSS Specifying this value is the same as omitting the PLSS keyword.

The default is NO_PLSS.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 1-byte field.

,SLSS=*slss*

,SLSS=NO_SLSS
An optional input parameter field containing the binary logical subsystem number within the secondary storage subsystem. This keyword is required for storage subsystems with logical subsystems and is invalid for 3990 and other 3990–compatible storage controls.

NO_SLSS Specifying this value is the same as omitting the SLSS keyword.

The default is NO_SLSS.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 1-byte field.

,OPENDVCS=*opendvcs*
,OPENDVCS=NO
An optional input parameter field containing a keyword that indicates if the request is for Open System (Fixed Block) devices. The value is left-justified and padded on the right with blanks. If the field contains binary zeros, the OPENDVCS keyword is treated as omitted.

YES Specifying this value indicates the information for the primary and secondary devices (SSID, Serial, LSS, and CCA) applies to Open System devices, not the CKD DEVN device. The CCA value is treated as a Fixed Block Logical Unit Number (LUN). The DEVN parameter identifies a CKD device for receiving the request, but is not affected by this PPRC request.

NO Specifying this value is the same as omitting the OPENDVCS keyword. This value indicates the DEVN device and primary and secondary devices are CKD devices.

The default is NO.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 3-character field.

,ECB=*ecb*
,ECB=NO_ECB
An optional input parameter field that SDM will post for an asynchronous request. This field is ignored for synchronous requests. The default is NO_ECB.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,WAITTIME=*waittime* **| 0**
An optional input parameter field containing how long in seconds SDM waits for a request to complete. A value of zero indicates that the system does not time the request. If the time expires before the request is complete a value of 7140 (RQST_WAITTIME_EXPIRED with a message ANTP0217E indicating I/O timeout) is returned. If the request is an asynchronous request, the value is used as the ECB post code. If the waittime is less than five seconds, then SDM changes waittime to five seconds. If the waittime is greater than 255 seconds, SDM changes waittime to 255 seconds.

,SUBCHSET=*subchset*
,SUBCHSET=0
Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is currently supported by the host system's processor and configured for the device.

If you specify a value of X'00' or omit this keyword, the command is issued to the device that is currently logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

To code: Specify the RS-type name, or address in register (2)-(12), of an optional 1-byte (character) field.

Subparameters for REQUEST=PDELPATH

REQUEST=PDELPATH
Asks for the CDELPATH function of PPRC to be executed.

,DEVN=*devn*
A required input parameter field containing the binary device number to use for I/O.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-byte field.

,PSERIAL=*pserialno*
An optional input parameter field containing the EBCDIC serial number of the primary storage subsystem. This parameter field contains 12 characters. The serial number must be right justified in the field, and padded on the left with character zeros as needed. This parameter is required if the **PWWNN** parameter is not specified. If this parameter field is set to binary zero, the **PSERIAL** parameter is treated as omitted.

If the PSERIAL parameter is specified, the SSERIAL parameter must be specified. If the PSERIAL and SSERIAL parameters are specified, the PWWNN and SWWNN parameters cannot be specified.

To code: Specify the RS-type address, or address in register (2)-(12), of a 12-character field.

,PWWNN=*pwwnn*
An optional input parameter field containing the WWNN of the primary storage subsystem. This parameter field is 8 bytes in length. This parameter is required if the **PSERIAL** parameter is not specified. If this parameter field is set to binary zero, the **PWWNN** parameter is treated as omitted.

If the PWWNN parameter is specified, the SWWNN parameter must be specified. If the PWWNN and SWWNN parameters are specified, the PSERIAL and SSERIAL parameters cannot be specified.

To code: Specify the RS-type address, or address in register (2)-(12), of a 12-character field.

,PSSID=*pssid*
A required input parameter field containing the binary subsystem ID of the primary storage subsystem. When removing a path between open device logical subsystems, this value should be specified as X'FFFF'.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-byte field.

,SSERIAL=*sserialno*
An optional parameter field containing the EBCDIC serial number of the recovery storage control. This parameter field contains 12 characters. The serial number must be right justified in the field, and padded on the left with character zeros as needed. This parameter is required if the SWWNN

parameter is not specified. If this parameter field is set to binary zero, the **SSERIAL** parameter is treated as omitted.

If the SSERIAL parameter is specified, the PSERIAL parameter must be specified. If the PSERIAL and SSERIAL parameters are specified, PWWNN and SWWNN parameters cannot be specified.

To code: Specify the RS-type address, or address in register (2)-(12), of a 12-character field.

,SWWNN=*swwnn*

An optional input parameter field containing the WWNN of the recovery storage control. This parameter field is 8 bytes in length. This parameter is required if the **SSERIAL** parameter is not specified. If this parameter field is set to binary zero, the **SWWNN** parameter is treated as omitted.

If the SWWNN parameter is specified, the PWWNN parameter must be specified. If the PWWNN and SWWNN parameters are specified, the PSERIAL and SSERIAL parameters cannot be specified.

To code: Specify the RS-type address, or address in register (2)-(12), of a 12-character field.

,SSSID=*sssid*

A required input parameter field containing the binary subsystem ID of the secondary storage control. When removing a path between open device logical subsystems, this value should be specified as X'FFFF'.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-byte field.

,RETINFO=*retinfo*

A required output parameter field that is used to return detailed information about the results of *executing* the request. Information about the results of *scheduling* the request are returned in the RETCODE and RSNCODE fields. Align the area on a word boundary. The first 4-bytes contain the return code and the second 4-bytes contain the reason code. The remainder of the information is dependent on the return and reason codes. See the coding example “SDM API usability guide” on page 575 about how to coordinate the RETCODE, RSNCODE, and RETINFO fields.

To code: Specify the RS-type address, or address in register (2)-(12), of a 100-character field.

,PLSS=*plss*
,PLSS=NO_PLSS

An optional input parameter field containing the binary logical subsystem number within the primary storage subsystem. This keyword is required for storage subsystems with logical subsystems and is invalid for 3990 and other 3990–compatible storage controls.

NO_PLSS Specifying this value is the same as omitting the PLSS keyword.

The default is NO_PLSS.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 1-byte field.

,SLSS=*slss*
,SLSS=NO_SLSS

An optional input parameter field containing the binary logical subsystem number within the secondary storage subsystem. This keyword is required for storage subsystems with logical subsystems and is invalid for 3990 and other 3990–compatible storage controls.

NO_SLSS Specifying this value is the same as omitting the SLSS keyword.

The default is NO_SLSS.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 1-byte field.

,ECB=*ecb* **|** **NO_ECB**

An optional input parameter field that SDM will post for an asynchronous request. This field is ignored for synchronous requests. The default is NO_ECB.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,WAITTIME=*waittime*
,WAITTIME=0

An optional input field containing how long in seconds SDM waits for a request to complete. A value of zero indicates that the system does not time the request. If the time expires before the request is complete, a value of 7160 (RQST_PDELPATH_ERROR, with a message ANTP0217E indicating I/O timeout) is returned. If the request is a synchronous request the value is placed in the return code part of RETINFO. If the request is an asynchronous request the value is used as the ECB post code. If waittime is less than five seconds, then SDM changes waittime to five seconds. If waittime is greater than 255 seconds, SDM changes waittime to 255 seconds.

,SUBCHSET=*subchset*
,SUBCHSET=0

Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is currently supported by the host system's processor and configured for the device.

If you specify a value of X'00' or omit this keyword, the command is issued to the device that is currently logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

To code: Specify the RS-type name, or address in register (2)-(12), of an optional 1-byte (character) field.

Subparameters for REQUEST=PESTPAIR

REQUEST=PESTPAIR

Asks for the CESTPAIR function of PPRC to be executed.

,DEVN=*devn*

A required input parameter field containing the binary device number to use for I/O.

If OPENDVCS=YES is specified, then *devn* identifies a System z device to act as a CKD access volume. This devicemust be located in the same subsystem cluster as the primary device.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-byte field.

,CASCADE=*cascade*
,CASCADE=YES | NO

Specifies whether primary volume in this pair is also eligible to be the secondary volume in another PPRC pair. The values are:

NO The primary volume is not eligible to be secondary in another PPRC pair. The default is NO.

YES
The primary volume is eligible to be secondary in another PPRC pair.

FORCE=*force*
FORCE=YES | NO

Specifies whether validation of the volumes involved in the establish request occurs, or is bypassed.

Note: This parameter is only valid when specified with MODE=INCRES, ACTION=FAILOVER, or ACTION=FAILBACK.

YES

- Specifying FORCE(YES) with MODE(INCRES) indicates that normal validation that the volumes in the request have an intermediate volume in common is bypassed.
- Specifying FORCE(YES) with ACTION(FAILOVER) or ACTION(FAILBACK) indicates that validation that the specified secondary is the existing suspended device in the existing relationship for the specified primary in FAILOVER mode is bypassed.

NO

- Specifying FORCE(NO) with MODE(INCRES) indicates the following validation:
  - The current failed secondary device for the primary specified, and the current failed primary device for the secondary specified are the same device.
- Specifying FORCE(NO) with ACTION(FAILOVER) or ACTION(FAILBACK) indicates the following validation:
  - The devices specified in the establish request are currently part of a suspended relationship (in failover state for ACTION(FAILBACK)).

To code: Specify the RS-type address, or address in register (2)-(12), of a 4-character field.

AUTORESYNC=*autoresync*
AUTORESYNC=YES|NO

Specifies the name (RS-type) or address in register (2) - (12) of a 3-character field that contains 'YES' or 'NO', specifying whether the auto resync capability is to be enabled or disabled for the Global Copy pair. The auto resync capability attempts to automatically resynchronize the pair if it has been suspended due to something other than a command. The specified character value is left-justified and padded on the right with blanks. The values are:

YES
Indicates that the auto resync capability should be enabled for the Global Copy pair. This is the default.

NO Indicates that the auto resync capability should be disabled for the Global Copy pair.

You may want to use NO to disable the auto resync capability when the secondary of the Global Copy pair is or will be the source of a Space Efficient FlashCopy relationship. An outage that results in many updates that must be copied when the pair is resynchronized could cause the space efficient repository to reach warning levels or even be filled.

If the keyword is not specified, the auto resync capability is enabled for the Global Copy pair. If NO is specified and option XD is not specified, the AUTORESYNC option is ignored.

INCRESYNC=*incresync*
Specifies whether the incremental re-synchronous change recording mechanism for M/GM is enabled. There is no default value for INCRESYNC. INCRESYNC is mutually exclusive with ACTION=FAILOVER.

START
The change recording mechanism is started.

STOP
The change recording mechanism is stopped.

NOCOPY
Used in recovery scenarios, where the incremental resync change recording mechanism needs to be started.

Note: If the INCRESYNC=NOCOPY parameter is specified in scenarios other than where it is intended, it cause you to make the recovery environment vulnerable to data integrity exposures. If blanks or binary zeros are specified, the INCRESYNC parameter is ignored.

,PCCA=*pcca*
A required input parameter field containing the binary channel connection address of the primary device. For open (fixed block) devices, this specifies the logical unit number (lun).

To code: Specify the RS-type address, or address in register (2)-(12), of a 1-byte field.

,PSERIAL=*pserial*
A required input parameter field containing the EBCDIC serial number of the primary storage control.

To code: Specify the RS-type address, or address in register (2)-(12), of a 12-character field.

,PSSID=*pssid*
A required input parameter field containing the binary subsystem ID of the primary storage control. When OPENDVCS=YES this value should be specified as X'FFFF'.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-byte field.

,SCCA=*scca*
A required input parameter field containing the binary channel connection address of the secondary device. For open (fixed block) devices, this specifies the logical unit number (lun).

To code: Specify the RS-type address, or address in register (2)-(12), of a 1-byte field.

,SSERIAL=*sserial*
A required input parameter field containing the EBCDIC serial number of the secondary storage control.

To code: Specify the RS-type address, or address in register (2)-(12), of a 12-character field.

,SSSID=*sssid*
A required input parameter field containing the binary subsystem ID of the secondary storage control. When OPENDVCS=YES this value should be specified as X'FFFF'.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-byte field.

,RETINFO=*retinfo*
A required output parameter field that is used to return detailed information about the results of *executing* the request. Information about the results of *scheduling* the request are returned in the RETCODE and RSNCODE fields. Align the area on a word boundary. The first 4-bytes contain the return code and the second 4-bytes contain the reason code. The remainder of the information is dependent on the return and reason codes. See the coding example “SDM API usability guide” on page 575 about how to coordinate the RETCODE, RSNCODE, and RETINFO fields.

To code: Specify the RS-type address, or address in register (2)-(12), of a 100-character field.

,PLSS=*plss*
,PLSS=NO_PLSS
An optional input parameter field containing the binary logical subsystem number within the primary storage subsystem. This keyword is required for storage subsystems with logical subsystems and is invalid for 3990 and other 3990–compatible storage controls.

NO_PLSS Specifying this value is the same as omitting the PLSS keyword.

The default is NO_PLSS.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 1-byte field.

,SLSS=*slss*
,SLSS=NO_SLSS
An optional input parameter field containing the binary logical subsystem number within the secondary storage subsystem. This keyword is required for storage subsystems with logical subsystems and is invalid for 3990 and other 3990–compatible storage controls.

NO_SLSS Specifying this value is the same as omitting the SLSS keyword.

The default is NO_SLSS.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 1-byte field.

,CRIT=*crit*
,CRIT=NO
An optional input parameter field containing a keyword that indicates if

this is a critical volume. The value is left-justified and padded on the right with blanks. If the field contains binary zeros, the program uses the default.

NO specifies that, following an I/O completion error, subsequent write requests to the PPRC primary volume are allowed.

YES specifies that if an I/O error occurs, subsequent writes are either allowed or not allowed, based on the 3990 maintenance panel setting. CRIT=YES is not supported with the ACTION parameter.

The default is NO.

To code: Specify the RS-type address, or address in register (2)-(12), of a 3-character field.

,MODE=*mode*
,MODE=COPY

An optional input parameter field containing a keyword that specifies how a volume pair is to be setup. The value is left-justified and padded on the right with blanks. If the field contains binary zeros, the program uses the default.

COPY specifies that all tracks on the primary volume are to be copied to the secondary volume. MODE=COPY is not supported with the ACTION parameter.

NOCOPY specifies that only those tracks on the primary volume that are updated after this request is executed are to be copied to the secondary volume.

INCRES causes that an incremental resynchronization using the change recording bitmaps to limit the tracks to be copied.

Note: MODE=INCRES is mutually exclusive with the all values for the ACTION and INCRESYNC keywords.

RESYNC specifies that a suspended copy be reestablished.

The default is COPY.

To code: Specify the RS-type address, or address in register (2)-(12), of a 6-character field.

,MSGREQ=*msgreq*
,MSGREQ=NO

An optional input parameter field containing a keyword that specifies whether to wait for FlashCopy Establish initialization to complete. The value is left-justified and padded on the right with blanks. If the field contains binary zeros, the program uses the default.

NO indicates to not wait.

YES indicates to wait.

The default is NO. MSGREQ is not supported with the ACTION parameter.

To code: Specify the RS-type address, or address in register (2)-(12), of a 3-character field.

,ONLINSEC=*onlinsec*
,ONLINSEC=NO

An optional input parameter field that contains a keyword that specifies whether Establish Pair processing should continue if the secondary device is possibly online to some host system. The value is left-justified and padded on the right with blanks. If the field contains binary zeros, the program uses the default.

YES indicates to continue Establish Pair processing if the secondary device is online to a host system.

NO indicates to fail Establish Pair processing if the secondary device is online to a host system.

The default is NO.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 3-character field.

,PACE=*pace*
,PACE=DEFAULT

An optional input parameter field that specifies the minimum number of tracks to be copied at one time before allowing another host interrupt. If a value is not supplied the default setup in the storage control will be used. The default is DEFAULT. PACE is not supported with the ACTION parameter.

To code: Specify the RS-type address, or address in register (2)-(12), of a 1-byte field.

,OPENDVCS=*opendvcs*
,OPENDVCS=NO

An optional input parameter field containing a keyword that indicates if the request is for Open System (Fixed Block) devices. The value is left-justified and padded on the right with blanks. If the field contains binary zeros, the OPENDVCS keyword is treated as omitted.

YES Specifying this value indicates the information for the primary and secondary devices (SSID, Serial, LSS, and CCA) applies to Open System devices, not the CKD DEVN device. The CCA value is treated as a Fixed Block Logical Unit Number (LUN). The DEVN parameter identifies a CKD device for receiving the request, but is not affected by this PPRC request.

NO Specifying this value is the same as omitting the OPENDVCS keyword. This value indicates the DEVN device and primary and secondary devices are CKD devices.

The default is NO.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 3-character field.

,ACTION=*action*

This field is the name (RS-type) (or address in register (2)-(12) of an optional 12-character field that identifies an action to take place on an already existing PPRC pair. The value is left justified and padded on the right with blanks. Values are:

FAILOVER

Specifies that the direction of the PPRC pair is to be reversed and change recording it to begin. Upon completion of the command, the specified primary device will be a suspended primary of a PPRC pair. Be sure to reverse devices when specifying primary and secondary parameters.

FAILBACK
Specifies that the failure condition at the primary site has been resolved and the changes that have taken place at the recovery site since the FAILOVER command was issued should now be copied to the original primary site. Be sure to specify original (non-reversed) primary and secondary device parameters.

MTFAILOVER
Specifies that a multi-target configuration should be created from a cascaded configuration. For example, in a cascaded configuration, H1->H2->H3, where H1 is the primary, issuing a CESTPAIR command with ACTION(MTFAILOVER) to the middle device, H2, causes a multi-target configuration (H2->H1 and H2-H3) in place of the previous cascaded failover state.

If ACTION is specified, the CRIT, CASCADE, PACE, and MSGREQ parameters are ignored.

SETGTOK=*setgtok***, SETGTOK=NO**
This parameter is the name (RS-type) or address in register (2)-(12) of a three-character field that specifies whether the secondary of the specified pair can be a space-efficient volume. The specified character value is left-justified and padded on the right with blanks.

NO Indicates that secondary cannot be a space-efficient volume. The default value is NO.

YES
Indicates that the secondary can be a space-efficient volume. If YES is specified and the secondary is not a space-efficient volume, the command is processed as if the keyword was not specified.

,ECB=*ecb*
,ECB=NO_ECB
An optional input parameter field that SDM will post for an asynchronous request. This field is ignored for synchronous requests. The default is NO_ECB.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,WAITTIME=*waittime*
,WAITTIME=0
An optional input field containing how long in seconds SDM waits for a request to complete. A value of zero indicates that the system does not time the request. If the time expires before the request is complete a value of, 7039, RQST_WAITTIME_EXPIRED (see ANTRQSTL) or 7200 (RQST_PESTPAIR_ERROR, with a message ANTP0217E indicating I/O timeout) is returned. If the request is a synchronous request, the value is placed in the return code part of RETINFO. If the request is an asynchronous request, the value is used as the ECB post code. Note that if waittime is specified with MSGREQ yes, then waittime greater than 255 seconds will not be reduced. Otherwise, if waittime is less than five seconds and MSGREQ is not yes, then SDM changes waittime to five seconds. If waittime is greater than 255 seconds, SDM changes waittime to 255 seconds.

,SUBCHSET=*subchset*
,SUBCHSET=0
Specifies the subchannel set in which the command is to be issued. The

subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is currently supported by the host system's processor and configured for the device.

If you specify a value of X'00' or omit this keyword, the command is issued to the device that is currently logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

To code: Specify the RS-type name, or address in register (2)-(12), of an optional 1-byte (character) field.

RESETRSV=*resetrsv***, RESETRSV=NO**

This parameter is the name (RS-type) or address in register (2)-(12) of a three-character field that specifies whether an existing reserve on the specified fixed block secondary should be reset (removed) by the control unit during the establish-pair processing. The specified character value is left-justified and padded on the right with blanks. The values are:

NO Indicates that any reserve that exists on the specified secondary device should not be reset. This is the default.

YES

Indicates that if there is a reserve on the specified secondary device, that reserve should be reset.

Subparameters for REQUEST=PESTPATH

REQUEST=PESTPATH

Asks for the CESTPATH function of PPRC to be executed.

,DEVN=*devn*

A required input parameter field containing the binary device number to use for I/O.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 2-character input field containing the hexadecimal device number to use for I/O.

The following is a set of mutually exclusive keys. This set is required; only one key must be specified.

,LINKS=*links*

Belongs to a set of mutually exclusive keys. A required variable parameter field aligned on a fullword and containing from 1 to 8 binary path address(es) from the application site storage control to the recovery site storage control. This field is structured as follows:

- A 4-byte field with a binary count of the path addresses in the following list. The path address count field must have a value from 1 through 8.
- A 4-byte reserved field.
- A list of 8-byte path address fields. This list can have from 1 to 8 path address fields. The path address fields have the following format.
  - A 2-byte length field that has the length of the path address. The value in the length field must be 4(X'0004').
  - A 6-byte path address area that has the path address. The path address is left-justified in the path address area, with the right-most 2-bytes unused and reserved.

To code: Specify the RS-type address, or address in register (2)-(12), of a variable- character field.

,LINK1=*link1*
Belongs to a set of mutually exclusive keys. A required input parameter field containing a binary path address from the application site storage control to the recovery site storage control.

To code: Specify the RS-type address, or address in register (2)-(12), of a 4-byte field.

,LINK2=*link2*
,LINK2=0
When LINK1=*link1* is specified, an optional input parameter field containing a binary path address from the application site storage control to the recovery site storage control. If the field contains binary zeros, the program uses the default. The default is 0.

To code: Specify the RS-type address, or address in register (2)-(12), of a 4-byte field.

,LINK3=*link3*
,LINK3=0
When LINK2=*link2* and LINK1=*link1* are specified, an optional input parameter field containing a binary path address from the application site storage control to the recovery site storage control. If the field contains binary zeros, the program uses the default. The default is 0.

To code: Specify the RS-type address, or address in register (2)-(12), of a 4-byte field.

,LINK4=*link4*
,LINK4=0
When LINK3=*link3*, LINK2=*link2*, and LINK1=*link1* are specified, an optional input parameter field containing a binary path address from the application site storage control to the recovery site storage control. If the field contains binary zeros, the program uses the default. The default is 0.

To code: Specify the RS-type address, or address in register (2)-(12), of a 4-byte field.

End of a set of mutually exclusive keys.

,PSERIAL=*pserialno*
An optional input parameter field containing the EBCDIC serial number of the primary storage subsystem. This parameter field contains 12 characters. The serial number must be right justified in the field, and padded on the left with character zeros as needed. This parameter is required if the **PWWNN** parameter is not specified. If this parameter field is set to binary zero, the **PSERIAL** parameter is treated as omitted.

When this parameter is used, the LINK addresses must have ESCON channel interface identifiers.

If the PSERIAL parameter is specified, the SSERIAL parameter must be specified. If the PSERIAL and SSERIAL parameters are specified, the PWWNN and SWWNN parameters cannot be specified.

To code: Specify the RS-type address, or address in register (2)-(12), of a 12-character field.

,PWWNN=*pwwnn*
An optional input parameter field containing the WWNN of the primary

storage subsystem. This parameter field is 8 bytes in length. This parameter is required if the **PSERIAL** parameter is not specified. If this parameter field is set to binary zero, the **PWWNN** parameter is treated as omitted.

When this parameter is used, the LINK addresses must have FCP adapter interface identifiers.

If the PWWNN parameter is specified, the SWWNN parameter must be specified. If the PWWNN and SWWNN parameters are specified, the PSERIAL and SSERIAL parameters cannot be specified.

To code: Specify the RS-type address, or address in register (2)-(12), of a 12-character field.

,PSSID=*pssid*

A required input parameter field containing the binary subsystem ID of the primary storage control. When establishing a path between open device logical subsystems, this value should be specified as X'FFFF'.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-byte field.

,SSERIAL=*sserialno*

An optional parameter field containing the EBCDIC serial number of the recovery storage control. This parameter field contains 12 characters. The serial number must be right justified in the field, and padded on the left with character zeros as needed. This parameter is required if the SWWNN parameter is not specified. If this parameter is set to binary zero, the SSERIAL parameter is treated as omitted.

When this parameter is used, the LINK addresses must have ESCON channel interface identifiers.

If the SSERIAL parameter is specified, the PSERIAL parameter must be specified. If the PSERIAL and SSERIAL parameters are specified, PWWNN and SWWNN parameters cannot be specified.

To code: Specify the RS-type address, or address in register (2)-(12), of a 12-character field.

,SWWNN=*swwnn*

An optional input parameter field containing the WWNN of the recovery storage control. This parameter field is 8 bytes in length. This parameter is required if the **SSERIAL** parameter is not specified. If this parameter field is set to binary zero, the **SWWNN** parameter is treated as omitted.

When this parameter is used, the LINK addresses must have FCP adapter interface identifiers.

If the SWWNN parameter is specified, the PWWNN parameter must be specified. If the PWWNN and SWWNN parameters are specified, the PSERIAL and SSERIAL parameters cannot be specified.

To code: Specify the RS-type address, or address in register (2)-(12), of a 12-character field.

,SSSID=*sssid*

A required input parameter field containing the binary subsystem ID of the secondary storage control. When establishing a path between open device logical subsystems, this value should be specified as X'FFFF'.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-byte field.

,PLSS=*plss*,PLSS=NO_PLSS
An optional input parameter field containing the binary logical subsystem number within the primary storage subsystem. This keyword is required for storage subsystems with logical subsystems and is invalid for 3990 and other 3990–compatible storage controls.

NO_PLSS Specifying this value is the same as omitting the PLSS keyword.

The default is NO_PLSS.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 1-byte field.

,SLSS=*slss*,SLSS=NO_SLSS
An optional input parameter field containing the binary logical subsystem number within the secondary storage subsystem. This keyword is required for storage subsystems with logical subsystems and is invalid for 3990 and other 3990–compatible storage controls.

NO_SLSS Specifying this value is the same as omitting the SLSS keyword.

The default is NO_SLSS.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 1-byte field.

,RETINFO=*retinfo*
A required output parameter field that is used to return detailed information about the results of *executing* the request. Information about the results of *scheduling* the request are returned in the RETCODE and RSNCODE fields. Align the area on a word boundary. The first 4-bytes contain the return code and the second 4-bytes contain the reason code. The remainder of the information is dependent on the return and reason codes. See the coding example "SDM API usability guide" on page 575 about how to coordinate the RETCODE, RSNCODE, and RETINFO fields.

To code: Specify the RS-type address, or address in register (2)-(12), of a 100-character field.

,CGROUP=*cgroup*
,CGROUP=NO
An optional input parameter field containing an indicator for logical control unit subsystems to emulate 3990 Critical Volume mode for CGROUP automation. The value is left-justified and padded on the right with blanks. This keyword is ignored when the device is not in a logical control unit subsystem. If the field contains binary zeros, the program uses the default.

NO indicates do not emulate 3990 Critical Volume mode.

YES indicates emulate 3990 Critical Volume mode to permit CGROUP automation processing of PPRC pair errors.

The default is NO.

To code: Specify the RS-type address, or address in register (2)-(12), of a 3-character field.

,RESETHP=*resethp*
,RESETHP=NO
An optional input parameter field containing a keyword that specifies

whether to reset host path connections to the primary subsystem before attempting to Establish Paths. The value is left-justified and padded on the right with blanks. This keyword is ignored when the device is not in a logical control unit subsystem. **NOTE:** This keyword is also ignored when you have specified a *wwnn* on the PRIM and SEC keywords. If the field contains binary zeros, the program uses the default.

YES indicates that host paths to the primary subsystem on the specified link addresses should be reset to permit Establish Paths to use the link addresses for connections to the secondary subsystems.

NO indicates that host path connections to the primary subsystem, using the same link addresses as specified in the Establish Paths request, will no be reset. The Establish Paths request will return an error indicating some or all paths failed to be established.

The default is NO.

To code: Specify the RS-type address, or address in register (2)-(12), of a 3-character field.

,ECB=*ecb*
,ECB=NO_ECB

An optional input parameter field that SDM will post for an asynchronous request. This field is ignored for synchronous requests. The default is NO_ECB.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,WAITTIME=*waittime*
,WAITTIME=0

An optional input parameter field containing how long in seconds SDM will wait for a request to complete. A value of zero says to not time the request. If the time expires before the request is complete a value of RQST_WAITTIME_EXPIRED (see ANTRQSTL) will be returned. If the request is a synchronous request the value will be placed in the RTNCODE field. If the request is an asynchronous request the value will be used as the ECB post code. The default is 0.

To code: Specify the RS-type address, or address in register (2)-(12), of a halfword field.

,SUBCHSET=*subchset*
,SUBCHSET=0

Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is currently supported by the host system's processor and configured for the device.

If you specify a value of X'00' or omit this keyword, the command is issued to the device that is currently logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

To code: Specify the RS-type name, or address in register (2)-(12), of an optional 1-byte (character) field.

Subparameters for REQUEST=PFREEZE

REQUEST=PFREEZE
Asks for the CGROUP function of PPRC to be executed.

,DEVN=*devn*
A required input parameter field containing the binary device number to use for I/O.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-byte field.

,PSERIAL=*pserial*
A required input parameter field containing the EBCDIC serial number of the primary storage control.

To code: Specify the RS-type address, or address in register (2)-(12), of a 12-character field.

,PSSID=*pssid*
A required input parameter field containing the binary subsystem ID of the primary storage control. When stopping updates to open device logical subsystems, this value should be specified as X'FFFF'.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-byte field.

,SSERIAL=*sserial*
A required input parameter field containing the EBCDIC serial number of the secondary storage control.

To code: Specify the RS-type address, or address in register (2)-(12), of a 12-character field.

,SSSID=*sssid*
A required input parameter field containing the binary subsystem ID of the secondary storage control. When stopping updates to open device logical subsystems, this value should be specified as X'FFFF'.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-byte field.

,RETINFO=*retinfo*
A required output parameter field that is used to return detailed information about the results of *executing* the request. Information about the results of *scheduling* the request are returned in the RETCODE and RSNCODE fields. Align the area on a word boundary. The first 4-bytes contain the return code and the second 4-bytes contain the reason code. The remainder of the information is dependent on the return and reason codes. See the coding example “SDM API usability guide” on page 575 about how to coordinate the RETCODE, RSNCODE, and RETINFO fields.

To code: Specify the RS-type address, or address in register (2)-(12), of a 100-character field.

,PLSS=*plss***,PLSS=NO_PLSS**
An optional input parameter field containing the hexadecimal logical subsystem number within the primary storage subsystem. This keyword is required for storage subsystems with logical subsystems and is invalid for 3990 and other 3990–compatible storage controls.

NO_PLSS Specifying this value is the same as omitting the PLSS keyword.

The default is NO_PLSS.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 1-character field.

,SLSS=*slss*,SLSS=NO_SLSS
An optional input parameter field containing the binary logical subsystem number within the secondary storage subsystem. This keyword is required for storage subsystems with logical subsystems and is invalid for 3990 and other 3990–compatible storage controls.

NO_SLSS Specifying this value is the same as omitting the SLSS keyword.

The default is NO_SLSS.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 1-byte field.

,ECB=*ecb*
,ECB=NO_ECB
An optional input parameter field that SDM will post for an asynchronous request. This field is ignored for synchronous requests. The default is NO_ECB.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,WAITTIME=*waittime*
,WAITTIME=0
An optional input parameter field containing how long in seconds SDM will wait for a request to complete. A value of zero says to not time the request. If the time expires before the request is complete a value of RQST_WAITTIME_EXPIRED (see ANTRQSTL) will be returned. If the request is a synchronous request the value will be placed in the RTNCODE field. If the request is an asynchronous request the value will be used as the ECB post code. The default is 0.

To code: Specify the RS-type address, or address in register (2)-(12), of a halfword field.

,SUBCHSET=*subchset*
,SUBCHSET=0
Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is currently supported by the host system's processor and configured for the device.

If you specify a value of X'00' or omit this keyword, the command is issued to the device that is currently logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

To code: Specify the RS-type name, or address in register (2)-(12), of an optional 1-byte (character) field.

Subparameters for REQUEST=PQSCSTAT

REQUEST=PQSCSTAT
Asks for the PQSCSTAT function of PPRC to be executed.

,DEVN=*devn*
A required input parameter field containing the binary device number to

use for I/O. This device must be located in the same subsystem cluster as the device specified in the QRYLSS and QRYDVC parameters.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-byte field.

QRYLSS=*qrylss*

A required input parameter field that specifies the two-digit hexadecimal value for the logical subsystem (LSS).

To code: Specify the RS-type address, or address in register (2)-(12), of a required 1-character field.

,QRYSIZE=*qrysize*

A required input parameter field that tells how big the area is that QRYINFO refers to.

To code: Specify the RS-type address, or address in register (2)-(12), of a halfword field.

,QRYINFO=*qryinfo*

A required output parameter field to be used to return the result of the query. The returned data is mapped by mapping macro (DSECT) ANTPQSCS. If the field *is not* big enough to contain the complete result a return code of RQST_PQUERY_QRYSIZE_TOO_SMALL (see ANTRQSTL) is placed in the return code part of RETINFO, and the total amount of space required is placed in the reason code part of RETINFO. If the field *is* big enough to contain the complete result a return code of RQST_PQUERY_QRYSIZE_BIG ENOUGH (see ANTRQSTL) is placed in the return code part of RETINFO, and the total amount of space actually used is placed in the reason code part of RETINFO.

The optional ALET parameter may be used to specify the location of the QRYINFO field.

To code: Specify the RS-type address, or address in register (2)-(12), of a character field.

,RETINFO=*retinfo*

A required output parameter field that is used to return detailed information about the results of *executing* the request. Information about the results of *scheduling* the request are returned in the RETCODE and RSNCODE fields. Align the area on a word boundary. The first 4-bytes contain the return code and the second 4-bytes contain the reason code. The remainder of the information is dependent on the return and reason codes. For information about how to coordinate the RETCODE, RSNCODE and RETINFO fields, refer to the coding example in “SDM API usability guide” on page 575.

To code: Specify the RS-type address, or address in register (2)-(12), of a 100-character field.

,OPENDVCS=*opendvcs*
, OPENDVCS=NO

An optional input parameter field containing a keyword that indicates if the request is for Open System (Fixed Block) devices. The value is left-justified and padded on the right with blanks. If the field contains binary zeros, the OPENDVCS keyword is treated as omitted.

YES indicates the query applies to an Open System LSS or device.

NO is the same as omitting the OPENDVCS keyword. This value indicates the query applies to a CKD LSS or device.

The default is NO.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 3-character field.

,QRYDVC =*qrydvc*

An optional input parameter field containing the two-digit hexadecimal value of the channel connection address (CCA) for the device that is to be queried. This parameter is required if SCOPE=RANK is specified; otherwise it is ignored.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 1-character field.

,SCOPE=*score*
,SCOPE=LSS

An optional input parameter field that specifies the content of the PQSCSTAT output. The default is LSS.

LSS indicates that device and LSS summary information only is to be returned.

RANK indicates that device-specific rank information as well as the LSS and device summary information are to be returned.

The default is LSS.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,SUBCHSET=*subchset*
,SUBCHSET=0

Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is currently supported by the host system's processor and configured for the device.

If you specify a value of X'00' or omit this keyword, the command is issued to the device that is currently logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

To code: Specify the RS-type name, or address in register (2)-(12), of an optional 1-byte (character) field.

,ALET=*alet*
,ALET=0

An optional input parameter field that specifies the ALET value to use if the QRYINFO field has an ALET value different from the one associated with the caller. The default is 0.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,ECB=*ecb*
,ECB=NO_ECB

An optional input parameter field that SDM will post for an asynchronous request. This field is ignored for synchronous requests. The default is NO_ECB.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,WAITTIME=*waittime*
,WAITTIME=0
An optional input field containing how long in seconds SDM waits for a request to complete. A value of zero indicates that the system does not time the request. If the time expires before the request is complete, a value of 7240 (RQST_PQUERY_ERROR, with a message ANTP0217E indicating I/O timeout) is returned. If the request is a synchronous request, the value is placed in the return code part of RETINFO. If the request is an asynchronous request, the value is used as the ECB post code. If *waittime* is less than five seconds, then SDM changes *waittime* to five seconds. If *waittime* is greater than 255 seconds, SDM changes *waittime* to 255 seconds.

Subparameters for REQUEST=PQUERY

REQUEST=PQUERY
Asks for the CQUERY function of PPRC to be executed.

,DEVN=*devn*
A required input parameter field containing the binary device number to use for I/O. If OPENDVCS=YES is specified, then *devn* identifies a System z device to act as a CKD access volume. This device must be located in the same subsystem cluster as the primary device.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-byte field.

,QRYSIZE=*qrysize*
A required input parameter field that tells how big the area is that QRYINFO refers to.

To code: Specify the RS-type address, or address in register (2)-(12), of a halfword field.

QRYSERIAL=*qryserial*
When OPENDVCS=YES, this specifies the storage control serial number that can include up to 12 digits, depending on the type of storage control.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 12-character field.

QRYLSS=*qrylss*
When OPENDVCS=YES, this specifies the two-digit hexadecimal value for the logical subsystem (LSS) for the device (ESS only).

Note: The LSS number is required if the storage control supports logical subsystems (like the ESS) and not allowed if the storage control does not support logical subsystems.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 1-character field.

QRYDVC =*qrydvc*
When OPENDVCS=YES, this specifies the two-digit hexadecimal value for the logical unit number (lun) for open devices, or the channel connection address *cca*.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 1-character field.

,QRYINFO=_qryinfo_

A required output parameter field to be used to return the result of the query. If the field *is not* big enough to contain the complete result a return code of RQST_PQUERY_QRYSIZE_TOO_SMALL (see ANTRQSTL) is placed in the return code part of RETINFO, and the total amount of space required is placed in the reason code part of RETINFO. If the field *is* big enough to contain the complete result a return code of RQST_PQUERY_QRYSIZE_BIG ENOUGH (see ANTRQSTL) is placed in the return code part of RETINFO, and the total amount of space actually used is placed in the reason code part of RETINFO.

For PATHS(NO) message ANTP0091I is indicated on the first line of the report. The output of this message is the unformatted query report. For PATHS(YES) message ANTP0096I is indicated on the last line of the report. The output for this message is the unformatted query report.

The optional ALET parameter may be used to specify the location of the QRYINFO field.

To code: Specify the RS-type address, or address in register (2)-(12), of a character field.

,RETINFO=_retinfo_

A required output parameter field that is used to return detailed information about the results of *executing* the request. Information about the results of *scheduling* the request are returned in the RETCODE and RSNCODE fields. Align the area on a word boundary. The first 4-bytes contain the return code and the second 4-bytes contain the reason code. The remainder of the information is dependent on the return and reason codes. See the coding example “SDM API usability guide” on page 575 about how to coordinate the RETCODE, RSNCODE, and RETINFO fields.

To code: Specify the RS-type address, or address in register (2)-(12), of a 100-character field.

,FORMAT=_format_

,FORMAT=LONG|PQMAP

An optional input parameter field containing a keyword that specifies the content of the PQUERY output. The value is left-justified and padded on the right with blanks. If the field is set to binary zero, the program uses the default.

LONG indicates that all fields in the PQUERY output are filled with relevant information as available. See Chapter 15, “Managing Peer-to-Peer Remote Copy operations,” on page 321 for further information. You can interpret the volume report by using “Querying PPRC volumes” on page 330 and the paths report by using “Querying PPRC paths” on page 345.

PQMAP provides the capability to return the result of a ANTRQST REQUEST=PQUERY in a format mapped by Dsect ANTPQMAP. It is returned in an area of storage that can be mapped using the ANTPQMAP MACRO. PQMAP indicates that the data is not to be formatted. It will be returned in an area of storage that can be mapped using the ANTPQMAP macro.

LONG is the default format.

To code: Specify the RS-type address, or address in register (2)-(12), of a six-character field.

,ALET=_alet_

,ALET=0

An optional input parameter field that specifies the ALET value to use if the QRYINFO field has an ALET value different from the one associated with the caller. The default is 0.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,BITMAP=*bitmap*
,BITMAP=YES

An optional input parameter field containing a keyword that specifies whether to access the bitmap for a primary device when determining the percentage of synchronization for a pending or suspended volume pair. This field is used only if DEVN specifies a primary volume. The value is left-justified and padded on the right with blanks. If the field contains binary zeros, the program uses the default.

NO directs PPRC to not read the bitmap for the primary volume specified by DEVN.

YES directs PPRC to read the primary volume's bitmap.

The default is YES.

Hint: If a P command has been issued to this device, the I/O to read the bitmap will not complete until a PRUN command is issued to this device. Use the WAITTIME parameter to return to the program that executed ANTRQST.

To code: Specify the RS-type address, or address in register (2)-(12), of a 3-character field.

,PATHS=*paths*
,PATHS=NO
,PATHS=YES
,PATHS=LNK

An optional input parameter field containing a keyword that specifies whether to specify volume information or path information. The value is left-justified and padded on the right with blanks. If the field contains binary zeros, the program uses the default.

NO directs PPRC to display the status of the volume specified by DEVN.

YES directs PPRC to display all of the paths associated with the application site storage control, and the status of each path.

LNK directs PPRC to display all of the potential connectivity of the Fibre Channel ports in the storage control specified by the DEVN parameter to each system adapter port in the storage control specified by the SWWNN or SDEVN parameter. If a device specified by SDEVN is the secondary of an established PPRC pair, the query will fail. The exception to this rule is if the secondary device is also the primary device of another PPRC pair (cascading). Either SWWNN or SDEVN is required.

The default is NO.

To code: Specify the RS-type address, or address in register (2)-(12), of a 3-character field.

,ECB=*ecb*

,ECB=NO_ECB
An optional input parameter field that SDM will post for an asynchronous request. This field is ignored for synchronous requests. The default is NO_ECB.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,SDEVN=*devn*
An optional input field containing the device number of the secondary volume that PPRC is to query. The device number is the 4-digit hexadecimal address of the device to which the I/O operation is directed.

To code: Specify the RS-type address, or address in register (2)-(12), of a 4-byte field.

,SWWNN=*wwnn*
An optional input field containing the world wide node name of the secondary volume that PPRC is to query. The world wide node name is the 8-byte hexadecimal identifier of the ESS to which the I/O operation is directed.

To code: Specify the RS-type address, or address in register (2)-(12), of an 8-byte field.

,WAITTIME=*waittime*
,WAITTIME=0
An optional input field containing how long in seconds SDM waits for a request to complete. A value of zero indicates that the system does not time the request. If the time expires before the request is complete, a value of 7240 (RQST_PQUERY_ERROR, with a message ANTP0217E indicating I/O timeout) is returned. If the request is a synchronous request, the value is placed in the return code part of RETINFO. If the request is an asynchronous request, the value is used as the ECB post code. If *waittime* is less than five seconds, then SDM changes *waittime* to five seconds. If *waittime* is greater than 255 seconds, SDM changes *waittime* to 255 seconds.

,SUBCHSET=*subchset*
,SUBCHSET=0
Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is currently supported by the host system's processor and configured for the device.

If you specify a value of X'00' or omit this keyword, the command is issued to the device that is currently logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

To code: Specify the RS-type name, or address in register (2)-(12), of an optional 1-byte (character) field.

,SSUBCHSET=*ssubchset*
,SSUBCHSET=0
ssubchset is the name (RS-type) or address in register (2)-(12) of a 1-character field that specifies the subchannel set for the device specified with the SDEVN parameter. This is the subchannel set as defined in the Hardware Configuration Dialog (HCD). Valid values are determined by what is supported by the host system's processor and configured for the device.

If you omit this parameter, the command is issued to the device that is logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD. If SSUBCHSET is specified without keyword SDEVN, it is ignored.

| **,QUIESCEDIO=***quiescedio*
| **,QUIESCEDIO=YES**
| An optional input parameter field containing a keyword that indicates if the request will be re-queued in the case of a device-not-ready condition caused by the quiesce processing. The value is left-justified and padded on the right with blanks. If the field contains binary zeros, the default is used.

| **YES** requeues the request. This is the default.

| **NO** immediately posts and fails the request.

| **To code:** Specify the RS-type address, or address in register (2)-(12), of a 3-character field.

Subparameters for REQUEST=PRECOVER

REQUEST=PRECOVER
Asks for the CRECOVER function of PPRC to be executed.

,DEVN=*devn*
A required input parameter field containing the binary device number to use for I/O.

If OPENDVCS=YES is specified, then *devn* identifies a System z device to act as a CKD access volume. This device must be located in the same subsystem cluster as the secondary device.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-byte field.

,OPENDVCS=*opendvcs*
,OPENDVCS=NO
An optional input parameter field containing a keyword that indicates if the request is for Open System (Fixed Block) devices. The value is left-justified and padded on the right with blanks. If the field contains binary zeros, the OPENDVCS keyword is treated as omitted.

YES Specifying this value indicates the information for the primary and secondary devices (SSID, Serial, LSS, and CCA) applies to Open System devices, not the CKD DEVN device. The CCA value is treated as a Fixed Block Logical Unit Number (LUN). The DEVN parameter identifies a CKD device for receiving the request, but is not affected by this PPRC request.

NO Specifying this value is the same as omitting the OPENDVCS keyword. This value indicates the DEVN device and primary and secondary devices are CKD devices.

The default is NO.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 3-character field.

,PCCA=*pcca*
A required input parameter field containing the binary channel connection address of the primary device. For open (fixed block) devices, this specifies the logical unit number (lun).

To code: Specify the RS-type address, or address in register (2)-(12), of a 1-byte field.

,PSERIAL=*pserial*
A required input parameter field containing the EBCDIC serial number of the primary storage control.

To code: Specify the RS-type address, or address in register (2)-(12), of a 12-character field.

,PSSID=*pssid*
A required input parameter field containing the binary subsystem ID of the primary storage control. When OPENDVCS=YES this value should be specified as X'FFFF'.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-byte field.

,SCCA=*scca*
A required input parameter field containing the binary channel connection address of the secondary device. For open (fixed block) devices, this specifies the logical unit number (lun).

To code: Specify the RS-type address, or address in register (2)-(12), of a 1-byte field.

,SSERIAL=*sserial*
A required input parameter field containing the EBCDIC serial number of the secondary storage control.

To code: Specify the RS-type address, or address in register (2)-(12), of a 12-character field.

,SSSID=*sssid*
A required input parameter field containing the binary subsystem ID of the secondary storage control. When OPENDVCS=YES this value should be specified as X'FFFF'.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-byte field.

,RETINFO=*retinfo*
A required output parameter field that is used to return detailed information about the results of *executing* the request. Information about the results of *scheduling* the request are returned in the RETCODE and RSNCODE fields. Align the area on a word boundary. The first 4-bytes contain the return code and the second 4-bytes contain the reason code. The remainder of the information is dependent on the return and reason codes. See the coding example "SDM API usability guide" on page 575 about how to coordinate the RETCODE, RSNCODE, and RETINFO fields.

To code: Specify the RS-type address, or address in register (2)-(12), of a 100-character field.

,PLSS=*plss***,PLSS=NO_PLSS**
An optional input parameter field containing the hexadecimal logical subsystem number within the primary storage subsystem. This keyword is required for storage subsystems with logical subsystems and is invalid for 3990 and other 3990–compatible storage controls.

NO_PLSS Specifying this value is the same as omitting the PLSS keyword.

The default is NO_PLSS.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 1-character field.

,SLSS=*slss* **,SLSS=NO_SLSS**
An optional input parameter field containing the hexadecimal logical subsystem number within the secondary storage subsystem. This keyword is required for storage subsystems with logical subsystems and is invalid for 3990 and other 3990–compatible storage controls.

NO_SLSS Specifying this value is the same as omitting the SLSS keyword.

The default is NO_SLSS.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 1-character field.

,OLDVOLSER=*oldvolser*
,OLDVOLSER=NO_VOLSER
An optional input parameter field that specifies the old volume serial number. If NEWVOLSER is not specified, PPRC verifies the serial number and resets the volume to simplex state. If the field contains binary zeros, the program uses the default. The default is NO_VOLSER. This parameter is not supported for open (fixed block) volumes.

To code: Specify the RS-type address, or address in register (2)-(12), of a 6-byte field.

,NEWVOLSER=*newvolser*
,NEWVOLSER=NO_VOLSER
When OLDVOLSER=*oldvolser* is specified, an optional input parameter field that specifies the new volume serial number. PPRC verifies the old serial number, resets the volume to the simplex state, then replaces the old serial number with the new serial number. If the field contains binary zeros, the program uses the default. The default is NO_VOLSER. This parameter is not supported for open (fixed block) volumes.

To code: Specify the RS-type address, or address in register (2)-(12), of a 6-byte field.

,ECB=*ecb*
,ECB=NO_ECB
An optional input parameter field that SDM will post for an asynchronous request. This field is ignored for synchronous requests. The default is NO_ECB.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,WAITTIME=*waittime*
,WAITTIME=0
An optional input field containing how long in seconds SDM waits for a request to complete. A value of zero indicates that the system does not time the request. If the time expires before the request is complete, a value of 7220 (RQST_PRECOVER_ERROR, with a message ANTP0217E indicating I/O timeout) is returned. If the request is a synchronous request, the value is placed in the return code part of RETINFO. If the request is an asynchronous request, the value is used as the ECB post code. If waittime is less than five seconds, then SDM changes waittime to five seconds. If waittime is greater than 255 seconds, SDM changes waittime to 255 seconds.

,SUBCHSET=*subchset*

,SUBCHSET=0
Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is currently supported by the host system's processor and configured for the device.

If you specify a value of X'00' or omit this keyword, the command is issued to the device that is currently logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

To code: Specify the RS-type name, or address in register (2)-(12), of an optional 1-byte (character) field.

Subparameters for REQUEST=PRUN

REQUEST=PRUN
Asks for the CGROUP Run function of PPRC to be executed.

,DEVN=*devn*
A required input parameter field containing the binary device number to use for I/O.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-byte field.

,PSERIAL=*pserial*
A required input parameter field containing the EBCDIC serial number of the primary storage control.

To code: Specify the RS-type address, or address in register (2)-(12), of a 12-character field.

,PSSID=*pssid*
A required input parameter field containing the binary subsystem ID of the primary storage control. When resuming application I/O on open devices, this value should be specified as X'FFFF'.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-byte field.

,SSERIAL=*sserial*
A required input parameter field containing the EBCDIC serial number of the secondary storage control.

To code: Specify the RS-type address, or address in register (2)-(12), of a 12-character field.

,SSSID=*sssid*
A required input parameter field containing the binary subsystem ID of the secondary storage control. When resuming application I/O on open devices, this value should be specified as X'FFFF'.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-byte field.

,RETINFO=*retinfo*
A required output parameter field that is used to return detailed information about the results of *executing* the request. Information about the results of *scheduling* the request are returned in the RETCODE and RSNCODE fields. Align the area on a word boundary. The first 4-bytes contain the return code and the second 4-bytes contain the reason code.

The remainder of the information is dependent on the return and reason codes. See the coding example "SDM API usability guide" on page 575 about how to coordinate the RETCODE, RSNCODE, and RETINFO fields.

To code: Specify the RS-type address, or address in register (2)-(12), of a 100-character field.

,PLSS=*plss***,PLSS=NO_PLSS**

An optional input parameter field containing the binary logical subsystem number within the primary storage subsystem. This keyword is required for storage subsystems with logical subsystems and is invalid for 3990 and other 3990–compatible storage controls.

NO_PLSS Specifying this value is the same as omitting the PLSS keyword.

The default is NO_PLSS.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 1-byte field.

,SLSS=*slss***,SLSS=NO_SLSS**

An optional input parameter field containing the binary logical subsystem number within the secondary storage subsystem. This keyword is required for storage subsystems with logical subsystems and is invalid for 3990 and other 3990–compatible storage controls.

NO_SLSS Specifying this value is the same as omitting the SLSS keyword.

The default is NO_SLSS.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 1-byte field.

,ECB=*ecb*
,ECB=NO_ECB

An optional input parameter field that SDM will post for an asynchronous request. This field is ignored for synchronous requests. The default is NO_ECB.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,WAITTIME=*waittime*
,WAITTIME=0

An optional input field containing how long in seconds SDM will wait for a request to complete. A value of zero says do not time the request. If the time expires before the request is complete, a value of 7260 (RQST_PRUN_ERROR, with a message ANTP0217E indicating I/O timeout) will be returned. If the request is a synchronous request the value will be placed in the return code part of RETINFO. If the request is an asynchronous request the value will be used as the ECB post code. If waittime is less than five seconds, then SDM will change waittime to five seconds. If waittime is greater than 255 seconds, SDM will change waittime to 255 seconds.

,SUBCHSET=*subchset*
,SUBCHSET=0

Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is currently supported by the host system's processor and configured for the device.

If you specify a value of X'00' or omit this keyword, the command is issued to the device that is currently logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

To code: Specify the RS-type name, or address in register (2)-(12), of an optional 1-byte (character) field.

Subparameters for REQUEST=PSUSPEND

REQUEST=PSUSPEND
Asks for the CSUSPEND function of PPRC to be executed.

,DEVN=*devn*
A required input parameter field containing the binary device number to use for I/O.

If OPENDVCS(YES) is specified, then *devn* identifies a System z device to act as a CKD access volume. This device must be located in the same subsystem cluster as the device(s) being suspended (primary or secondary).

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-byte field.

,PRIMARY=*primary*
,PRIMARY=NO
An optional input parameter field containing a keyword that indicates ICKDSF media maintenance is wanted. The value is left-justified and padded on the right with blanks. If the field contains binary zeros, the program uses the default.

NO indicates that the request is a normal suspend.

YES directs PPRC to suspend the PPRC volume pair and to reject all write data I/O to the primary volume except for ICKDSF media maintenance channel programs.

The default is NO.

To code: Specify the RS-type address, or address in register (2)-(12), of a 3-character field.

,QUIESCE=*quiesce*
,QUIESCE=NO
An optional input parameter field containing a keyword that directs PPRC to suspend the PPRC volume and to send a “state change pending” indication to the system to prevent issuance of any application system I/O. The value is left-justified and padded on the right with blanks. If the field contains binary zeros, the program uses the default.

NO indicates that quiesce is not wanted.

YES indicates quiesce is wanted.

The default is NO.

To code: Specify the RS-type address, or address in register (2)-(12), of a 3-character field.

,PCCA=*pcca*
A required input parameter field containing the binary channel connection address of the primary device. For open (fixed block) devices, this specifies the logical unit number (lun).

To code: Specify the RS-type address, or address in register (2)-(12), of a 1-byte field.

,PSERIAL=*pserial*
A required input parameter field containing the EBCDIC serial number of the primary storage control.

To code: Specify the RS-type address, or address in register (2)-(12), of a 12-character field.

,PSSID=*pssid*
A required input parameter field containing the binary subsystem ID of the primary storage control. When OPENDVCS=YES, this value should be specified as X'FFFF'.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-byte field.

,SCCA=*scca*
A required input parameter field containing the binary channel connection address of the secondary device. For open (fixed block) devices, this specifies the logical unit number (lun).

To code: Specify the RS-type address, or address in register (2)-(12), of a 1-byte field.

,SSERIAL=*sserial*
A required input parameter field containing the EBCDIC serial number of the secondary storage control.

To code: Specify the RS-type address, or address in register (2)-(12), of a 12-character field.

,SSSID=*sssid*
A required input parameter field containing the binary subsystem ID of the secondary storage control. When OPENDVCS=YES, this value should be specified as X'FFFF'.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-byte field.

,RETINFO=*retinfo*
A required output parameter field that is used to return detailed information about the results of *executing* the request. Information about the results of *scheduling* the request are returned in the RETCODE and RSNCODE fields. Align the area on a word boundary. The first 4-bytes contain the return code and the second 4-bytes contain the reason code. The remainder of the information is dependent on the return and reason codes. See the coding example "SDM API usability guide" on page 575 about how to coordinate the RETCODE, RSNCODE, and RETINFO fields.

To code: Specify the RS-type address, or address in register (2)-(12), of a 100-character field.

,PLSS=*plss***,PLSS=NO_PLSS**
An optional input parameter field containing the binary logical subsystem number within the primary storage subsystem. This keyword is required for storage subsystems with logical subsystems and is invalid for 3990 and other 3990–compatible storage controls.

NO_PLSS Specifying this value is the same as omitting the PLSS keyword.

The default is NO_PLSS.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 1-byte field.

,SLSS=*slss***,SLSS=NO_SLSS**

An optional input parameter field containing the hexadecimal logical subsystem number within the secondary storage subsystem. This keyword is required for storage subsystems with logical subsystems and is invalid for 3990 and other 3990-compatible storage controls.

NO_SLSS Specifying this value is the same as omitting the SLSS keyword.

The default is NO_SLSS.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 1-character field.

,OPENDVCS=*opendvcs*
,OPENDVCS=NO

An optional input parameter field containing a keyword that indicates if the request is for Open System (Fixed Block) devices. The value is left-justified and padded on the right with blanks. If the field contains binary zeros, the OPENDVCS keyword is treated as omitted.

YES Specifying this value indicates the information for the primary and secondary devices (SSID, Serial, LSS, and CCA) applies to Open System devices, not the CKD DEVN device. The CCA value is treated as a Fixed Block Logical Unit Number (LUN). The DEVN parameter identifies a CKD device for receiving the request, but is not affected by this PPRC request.

NO Specifying this value is the same as omitting the OPENDVCS keyword. This value indicates the DEVN device and primary and secondary devices are CKD devices.

The default is NO.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 3-character field.

,ECB=*ecb*
,ECB=NO_ECB

An optional input parameter field that SDM will post for an asynchronous request. This field is ignored for synchronous requests. The default is NO_ECB.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,WAITTIME=*waittime*
,WAITTIME=0

An optional input field containing how long in seconds SDM waits for a request to complete. A value of zero indicates that the system does not time the request. If the time expires before the request is complete, a value of 7180 (RQST_PSUSPEND_ERROR, with a message ANTP0217E indicating I/O timeout) is returned. If the request is a synchronous request, the value is placed in the return code part of RETINFO. If the request is an asynchronous request, the value is used as the ECB post code. If waittime is less than five seconds, then SDM changes waittime to five seconds. If waittime is greater than 255 seconds, SDM changes waittime to 255 seconds.

,SUBCHSET=*subchset*

,SUBCHSET=0
Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is currently supported by the host system's processor and configured for the device.

If you specify a value of X'00' or omit this keyword, the command is issued to the device that is currently logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

To code: Specify the RS-type name, or address in register (2)-(12), of an optional 1-byte (character) field.

Subparameters for REQUEST=PSETCHAR

REQUEST=PSETCHAR
Asks for the PSETCHAR function of PPRC to be executed.

,DEVN=*devn*
A required input parameter field containing the binary device number topss use for I/O.

If OPENDVCS(YES) is specified, then *devn* identifies a System z device to act as a CKD access volume. This device must be located in the same subsystem cluster as the primary device.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-byte field.

,PCCA=*pcca*
A required input parameter field containing the binary channel connection address of the primary device. For open (fixed block) devices, this specifies the logical unit number (lun).

To code: Specify the RS-type address, or address in register (2)-(12), of a 1-byte field.

,PSERIAL=*pserial*
A required input parameter field containing the EBCDIC serial number of the primary storage control.

To code: Specify the RS-type address, or address in register (2)-(12), of a 12-character field.

,PLSS=*plss*
A required input parameter field containing the binary logical subsystem number within the primary storage subsystem.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 1-byte field.

,SCCA=*scca*
A required input parameter field containing the binary channel connection address of the secondary device. For open (fixed block) devices, this specifies the logical unit number (lun).

To code: Specify the RS-type address, or address in register (2)-(12), of a 1-byte field.

,SSERIAL=*sserial*
A required input parameter field containing the EBCDIC serial number of the secondary storage control.

To code: Specify the RS-type address, or address in register (2)-(12), of a 12-character field.

,SLSS=*slss*
A required input parameter field containing the binary logical subsystem number within the secondary storage subsystem.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 1-character field.

,USEFORPM=*useforpm*
,USEFORPM=NO
A required parameter field indicating whether the specified pair is to be used for a Preserve Mirror function in a Multi-Target Mirror configuration. One and only one pair from a PPRC primary can have this enabled or Preserve Mirror Required (Remote Pair FlashCopy) functions will fail due to ambiguous settings.

NO Specifies that this pair is not to be used for a Preserve Mirror function in a Multi-Target Mirror configuration.

YES Specifies that this pair is to be used for a Preserve Mirror function in a Multi-Target Mirror configuration.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 3-character field.

,RETINFO=*retinfo*
A required output parameter field that is used to return detailed information about the results of *scheduling* the request. Information about the results of *scheduling* the request are returned in the RETCODE and RSNCODE fields. Align the area on a word boundary. The first 4-bytes contain the return code and the second 4-bytes contain the reason code. The remainder of the information is dependent on the return and reason codes. See the coding example "SDM API usability guide" on page 575 about how to coordinate the RETCODE, RSNCODE, and RETINFO fields.

To code: Specify the RS-type address, or address in register (2)-(12), of a 100-character field.

,OPENDVCS=*opendvcs*
,OPENDVCS=NO
An optional input parameter field containing a keyword that indicates if the request is for open system (fixed block) devices. The value is left-justified and padded on the right with blanks. If the field contains binary zeros, the OPENDVCS keyword is treated as omitted.

YES specifies that the information for the primary and secondary devices (serial, LSS, and CCA) applies to open system devices, not the CKD DEVN device. The CCA value is treated as a fixed block logical unit number (LUN). The DEVN parameter identifies a CKD device for receiving the request, but is not affected by this PPRC request.

NO has the same effect as omitting the OPENDVCS keyword. This value indicates the DEVN device and primary and secondary devices are CKD devices.

The default is NO.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 3-character field.

,ECB=*ecb*
,ECB=NO_ECB
An optional input parameter field that SDM will post for an asynchronous request. This field is ignored for synchronous requests. The default is NO_ECB.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,WAITTIME=*waittime*
,WAITTIME=0
An optional input field containing how long in seconds SDM waits for a request to complete. A value of zero indicates that the system does not time the request. If the time expires before the request is complete a value of 7039, RQST_WAITTIME_EXPIRED (see ANTRQSTL) or 7200 (RQST_PESTPAIR_ERROR, with a message ANTP0217E indicating I/O timeout) is returned. If the request is a synchronous request, the value is placed in the return code part of RETINFO. If the request is an asynchronous request, the value is used as the ECB post code. Note that if WAITTIME is specified with MSGREQ YES, then a wait time of greater than 255 seconds will not be reduced. Otherwise, if *waittime* is less than five seconds and MSGREQ is not YES, then SDM changes *waittime* to five seconds. If *waittime* is greater than 255 seconds, SDM changes *waittime* to 255 seconds.

SUBCHSET=*subchset*
,SUBCHSET=0
subchset is the name (RS-type) or address in register (2)-(12) of a 1-character field that specifies the subchannel set for the device where the command is to be issued. This is the subchannel set as defined in the Hardware Configuration Dialog (HCD). Valid values are:

0 Subchannel set 0. This is the default.

1 Subchannel set 1

2 Subchannel set 2

3 Subchannel set 3.

If you specify a value of X'00' or omit this keyword, the command is issued to the default device that is logically in subchannel set 0. If a swap has occurred, this could be the device that was physically defined in an alternate subchannel set. This is the default.

Parameter descriptions for ILK=SNAPSHOT

When you invoke the ANTRQST macro for Snapshot, you can use the following REQUEST parameters. For a list of the Common End Codes associated with each request within this ILK, see “Common end codes for REQUESTS within each ILK” on page 735.

Subparameters for REQUEST=LEVEL

REQUEST=LEVEL
Asks for the level of ANTRQST that is installed on the system.

,RETINFO=*retinfo*
is the name (RS-type) (or address in register (2)-(12) ASM only) of a required 100-character output field used to return the level of ANTRQST installed on the system. If ANTRQST is not installed, return code 7000 (RQST_PC_NUMBER_ZERO - see ANTRQSTL) is returned in register 15 and placed in the RETCODE area, if specified. If ANTRQST is installed, a level value is placed in the first 4-bytes of the RETINFO field. This level value indicates which ANTRQST functions, keywords and parameters are supported. For more information about the LEVEL request output, see the prolog comments in the ANTRQSTL macro (shipped with the ANTRQST macro).

Subparameters for REQUEST=SDVCINFO

REQUEST=SDVCINFO
Asks for information on an RVA device.

,DEVN=*devn*
,VOLSER=*volser*
,UCBPTR=*ucbptr*
One of the following three input parameters is required:

,DEVN=*devn*
A parameter field containing the binary device number of the device you are interested in.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-byte field.

,VOLSER=*volser*
A parameter field containing the volume serial of the device you are interested in.

To code: Specify the RS-type address, or address in register (2)-(12), of a 6-byte field.

,UCBPTR=*ucbptr*
A parameter field containing the UCB address of the device you are interested in.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,BASICSIZE=*basicsize*
A required input field that indicates the size of the area that BASICINFO refers to. The minimum size that can be specified is the length of PSL10_HEADER (see BASICINFO).

To code: Specify the RS-type address of a halfword field, or register (2)-(12).

,BASICINFO=*basicinfo*
A required output parameter field that is used to contain the result of the query for the device you are interested in. The information is described in mapping macro ANTPSL10, which is included as part of Snapshot support. If the field is large enough to hold all of the data, PSL10_DATA_USED tells how much data was returned. If the field is not large enough, PSL10_DATA_TRUNC is turned on, PSL10_DATA_USED tells how much space was required, and a return code of RQST_SDVCINFO_BASICSIZE_TOO_SMALL (see ANTRQSTL) is placed in the return code part of RETINFO.

The optional ALET parameter can be used to specify the location of the BASICINFO field.

To code: Specify the RS-type address of a variable-length field, or register (2)-(12).

,RETINFO=*retinfo*
A required output parameter field that is used to return detailed information about the results of *executing* the request. Information about the results of *scheduling* the request are returned in the RETCODE and RSNCODE fields. Align the area on a word boundary. The first 4-bytes contain the return code and the second 4-bytes contain the reason code. The remainder of the information is dependent on the return and reason codes. See the coding example "SDM API usability guide" on page 575 about how to coordinate the RETCODE, RSNCODE, and RETINFO fields.

To code: Specify the RS-type address, or address in register (2)-(12), of a 100-character field.

,ALET=*alet*
,ALET=0
An optional input parameter field that specifies the ALET value to be used if the BASICINFO and EXTNDINFO fields have an ALET value different from the one associated with the caller. The default is 0.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,EXTNDSIZE=*extndsize*
,EXTNDSIZE=0
An optional input field that indicates the size of the area that EXTNDINFO refers to. The minimum size that can be specified is the length of PSL11_HEADER (see EXTNDINFO). The default is 0.

To code: Specify the RS-type address, or address in register (2)-(12), of a halfword field.

,EXTNDINFO=*extndinfo*
An optional output field used to contain the extended information for the device you are interested in. The information is described in mapping macro ANTPSL11, which is included as part of Snapshot support. If the field is large enough to hold all of the data, PSL11_DATA_USED tells how much data was returned. If the field is not large enough, PSL11_DATA_TRUNC is turned on, PSL11_DATA_USED tells how much space was required, and a return code of RQST_SDVCINFO_EXTNDSIZE_TOO_SMALL (see ANTRQSTL) is placed in the return code part of RETINFO.

The optional ALET parameter can be used to specify the location of the EXTNDINFO field.

To code: Specify the RS-type address, or address in register (2)-(12), of a variable-length field.

,ECB=*ecb*
,ECB=NO_ECB
An optional input parameter field that SDM will post for an asynchronous request. This field is ignored for synchronous requests. The default is NO_ECB.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,WAITTIME=*waittime*
,WAITTIME=0
An optional input parameter field containing how long in seconds SDM will wait for a request to complete. A value of zero says to not time the request. If the time expires before the request is complete a value of RQST_WAITTIME_EXPIRED (see ANTRQSTL) will be returned. If the request is a synchronous request the value will be placed in the RTNCODE field. If the request is an asynchronous request the value will be used as the ECB post code. The default is 0.

To code: Specify the RS-type address, or address in register (2)-(12), of a halfword field.

Subparameters for REQUEST=SQRYDVCS

REQUEST=SQRYDVCS
Asks for information on the RVA devices attached to a host.

,QRYSIZE=*qrysize*
A required input parameter field that tells how big the area is that QRYINFO refers to. The minimum size that can be specified is the length of PSL30_HEADER (see QRYINFO).

To code: Specify the RS-type address, or address in register (2)-(12), of a halfword field.

,QRYINFO=*qryinfo*
A required output parameter field to be used to return the result of the query. The returned information is described in mapping macro ANTPSL30, which is included as part of Snapshot support. If the field is large enough to hold all of the data PSL30_DATA_USED will tell how much data was returned. If the field was not large enough, PSL30_DATA_TRUNC will be turned on, PSL30_DATA_USED will tell how much space was required, and a return code of RQST_SQRYDVCS_QRYSIZE_TOO_SMALL (see ANTRQSTL) will be placed in the return code part of RETINFO.

The optional ALET parameter may be used to specify the location of the QRYINFO field.

To code: Specify the RS-type address, or address in register (2)-(12), of a character field.

,RETINFO=*retinfo*
A required output parameter field that is used to return detailed information about the results of *executing* the request. Information about the results of *scheduling* the request are returned in the RETCODE and RSNCODE fields. Align the area on a word boundary. The first 4-bytes contain the return code and the second 4-bytes contain the reason code. The remainder of the information is dependent on the return and reason codes. See the coding example “SDM API usability guide” on page 575 about how to coordinate the RETCODE, RSNCODE, and RETINFO fields.

To code: Specify the RS-type address, or address in register (2)-(12), of a 100-character field.

,ALET=*alet*
,ALET=0
An optional input parameter field that specifies the ALET value to be used if the QRYINFO field has an ALET value different from the one associated with the caller. The default is 0.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,SSFILTER=*ssfilter*
,SSFILTER=NO_FILTER
An optional input parameter field containing a filter to be used in determining what subsystems the devices being listed may come from. If this parameter is not specified, or the value in SSFSIZE is zero, subsystem names will not be considered when determining device eligibility. The rules for filter patterns are in item "Character filtering" on page 701. The default is NO_FILTER.

To code: Specify the RS-type address, or address in register (2)-(12), of a character field.

,SSFSIZE=*ssfsize*
When SSFILTER=*ssfilter* is specified, a required input parameter field that tells how big the area is that SSFILTER refers to. If the value in the field is zero filtering will not be done.

To code: Specify the RS-type address, or address in register (2)-(12), of a halfword field.

,VLFILTER=*vlfilter*
,VLFILTER=NO_FILTER
An optional input parameter field containing a filter to be used in determining what volumes the devices being listed may come from. If this parameter is not specified, or the value in VLFSIZE is zero, volume serial numbers will not be considered when determining device eligibility. The rules for filter patterns are in item "Character filtering" on page 701. The default is NO_FILTER.

To code: Specify the RS-type address, or address in register (2)-(12), of a character field.

,VLFSIZE=*vlfsize*
When VLFILTER=*vlfilter* is specified, a required input parameter field that tells how big the area is that VLFILTER refers to. If the value in the field is zero filtering will not be done.

To code: Specify the RS-type address, or address in register (2)-(12), of a halfword field.

,LOWDEVN=*lowdevn*
, LOWDEVN=NO_DEVN
An optional input parameter field containing the low binary device number of a range of devices you are interested in. If the field contains binary zeros, the program uses the default. The default is NO_DEVN.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-byte field.

,HIGHDEVN=*highdevn*
When LOWDEVN=*lowdevn* is specified, a required input parameter field. It contains the high binary device number of a range of devices you are interested in.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-byte field.

,DEVTYPE=*devtype*

,DEVTYPE=NO
An optional input parameter field containing a keyword that indicates whether a device type is to be used as a selection criteria. The value is left-justified and padded on the right with blanks. If the field contains binary zeros, the program uses the default.

NO indicates that device type is not to be considered.

3380 indicates that selected devices must have a device type of 3380.

3390 indicates that selected devices must have a device type of 3390.

The default is NO.

To code: Specify the RS-type address, or address in register (2)-(12), of a 4-character field.

,PARTITION=*partition*
,PARTITION=NO
An optional input parameter field containing a keyword that indicates whether a RVA partition is to be used as a selection criteria. The value is left-justified and padded on the right with blanks. If the field contains binary zeros, the program uses the default.

NO indicates that partition type is not to be considered.

TEST indicates that selected devices must be in a test partition.

PROD indicates that selected devices must be in a production partition.

The default is NO.

To code: Specify the RS-type address, or address in register (2)-(12), of a 4-character field.

,ACCESS=*access*
,ACCESS=NO
An optional input parameter field containing a keyword that indicates whether a read/write status is to be used as a selection criteria. The value is left-justified and padded on the right with blanks. If the field contains binary zeros, the program uses the default.

NO indicates that read/write status is not to be considered.

R indicates that selected devices must be read-only devices.

RW indicates that selected devices must be read/write devices.

The default is NO.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-character field.

,BEGINCYL0=*begincyl0*
,BEGINCYL0=YES
An optional input parameter field containing a keyword that indicates devices that do or do not begin on real cylinder 0 are to be included in the device list. The value is left-justified and padded on the right with blanks. If the field contains binary zeros, the program uses the default.

YES indicates that *only* devices that begin at real cylinder 0 are eligible to be in the list.

NO indicates that *all* devices, whether they begin at real cylinder 0 (e.g. VM minidisks), are eligible to be in the list.

The default is YES.

To code: Specify the RS-type address, or address in register (2)-(12), of a 3-character field.

,DEVN=*devn*
,DEVN=NO_DEVN
,VOLSER=*volser*
,UCBPTR=*ucbptr*
Only one of the following three optional input parameters may be coded:

,DEVN=*devn*
A parameter field containing the binary device number of a device. All devices on the same subsystem as the specified device are eligible to be in the list. The default is NO_DEVN.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-byte field.

,VOLSER=*volser*
A parameter field containing the volser of a device. All devices on the same subsystem as the specified device are eligible to be in the list.

To code: Specify the RS-type address, or address in register (2)-(12), of a 6-character field.

,UCBPTR=*ucbptr*
A parameter field containing the UCB address of for a device. All devices on the same subsystem as the specified device are eligible to be in the list.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,ECB=*ecb*
,ECB=NO_ECB
An optional input parameter field that SDM will post for an asynchronous request. This field is ignored for synchronous requests. The default is NO_ECB.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,WAITTIME=*waittime*
,WAITTIME=0
An optional input parameter field containing how long in seconds SDM will wait for a request to complete. A value of zero says to not time the request. If the time expires before the request is complete a value of RQST_WAITTIME_EXPIRED (see ANTRQSTL) will be returned. If the request is a synchronous request the value will be placed in the RTNCODE field. If the request is an asynchronous request the value will be used as the ECB post code. The default is 0.

To code: Specify the RS-type address, or address in register (2)-(12), of a halfword field.

Subparameters for REQUEST=SQRYSSYS

REQUEST=SQRYSSYS
Asks for a list of RVA subsystems attached to a host.

,QRYSIZE=*qrysize*

A required input parameter field that tells how big the area is that QRYINFO refers to. The minimum size that can be specified is the length of PSL20_HEADER (see QRYINFO).

To code: Specify the RS-type address, or address in register (2)-(12), of a halfword field.

,QRYINFO=*qryinfo*

A required output parameter field to be used to return the result of the query. The returned information is described in mapping macro ANTPSL20, which is included as part of Snapshot support. If the field is large enough to hold all of the data PSL20_DATA_USED will tell how much data was returned. If the field was not large enough, PSL20_DATA_TRUNC will be turned on, PSL20_DATA_USED will tell how much space was required, and a return code of RQST_SQRYDVCS_QRYSIZE_TOO_SMALL (see ANTRQSTL) will be placed in the return code part of RETINFO.

The optional ALET parameter may be used to specify the location of the QRYINFO field.

To code: Specify the RS-type address, or address in register (2)-(12), of a character field.

,RETINFO=*retinfo*

A required output parameter field that is used to return detailed information about the results of *executing* the request. Information about the results of *scheduling* the request are returned in the RETCODE and RSNCODE fields. Align the area on a word boundary. The first 4-bytes contain the return code and the second 4-bytes contain the reason code. The remainder of the information is dependent on the return and reason codes. See the coding example "SDM API usability guide" on page 575 about how to coordinate the RETCODE, RSNCODE, and RETINFO fields.

To code: Specify the RS-type address, or address in register (2)-(12), of a 100-character field.

,ALET=*alet*
,ALET=0

An optional input parameter field that specifies the ALET value to be used if the QRYINFO field has an ALET value different from the one associated with the caller. The default is 0.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,SSFILTER=*ssfilter*
,SSFILTER=NO_FILTER

An optional input parameter field containing a filter to be used in determining what subsystem names to return. If this parameter is not specified, or the value in SSFSIZE is zero, all subsystem names will be returned. The rules for filter patterns are in item "Character filtering" on page 701. The default is NO_FILTER.

To code: Specify the RS-type address, or address in register (2)-(12), of a character field.

,SSFSIZE=*ssfsize*
When SSFILTER=*ssfilter* is specified, a required input parameter field that tells how big the area is that SSFILTER refers to. If the value in the field is zero filtering will not be done.

To code: Specify the RS-type address, or address in register (2)-(12), of a halfword field.

,ECB=*ecb*
,ECB=NO_ECB
An optional input parameter field that SDM will post for an asynchronous request. This field is ignored for synchronous requests. The default is NO_ECB.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,WAITTIME=*waittime*
,WAITTIME=0
An optional input parameter field containing how long in seconds SDM will wait for a request to complete. A value of zero says to not time the request. If the time expires before the request is complete a value of RQST_WAITTIME_EXPIRED (see ANTRQSTL) will be returned. If the request is a synchronous request the value will be placed in the RTNCODE field. If the request is an asynchronous request the value will be used as the ECB post code. The default is 0.

To code: Specify the RS-type address, or address in register (2)-(12), of a halfword field.

Subparameters for REQUEST=SRELEASE

REQUEST=SRELEASE
is used to release functional tracks on an RVA device.

,DEVN=*devn*
,VOLSER=*volser*
,UCBPTR=*ucbptr*
One of the following three input parameters is required:

,DEVN=*devn*
A parameter field containing the binary device number of a device whose functional tracks are to be released.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-byte field.

,VOLSER=*volser*
A parameter field containing the volser of a device whose functional tracks are to be released.

To code: Specify the RS-type address, or address in register (2)-(12), of a 6-character field.

,UCBPTR=*ucbptr*
A parameter field containing the UCB address of for a device whose functional tracks are to be released.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,EXTINFO=*extinfo*
A required input parameter field that is an array of eight byte extent pairs

to release. The first 4-bytes of a pair gives the starting track address and the last 4-bytes gives the ending track address (for a single track the two addresses would be the same).

To code: Specify the RS-type address, or address in register (2)-(12), of a character field.

,EXTNBR=*extnbr*

A required input parameter field that tells how many extent pairs have been supplied in EXTINFO.

To code: Specify the RS-type address, or address in register (2)-(12), of a halfword field.

,RETINFO=*retinfo*

A required output parameter field that is used to return detailed information about the results of *executing* the request. Information about the results of *scheduling* the request are returned in the RETCODE and RSNCODE fields. Align the area on a word boundary. The first 4-bytes contain the return code and the second 4-bytes contain the reason code. The remainder of the information is dependent on the return and reason codes. See the coding example “SDM API usability guide” on page 575 about how to coordinate the RETCODE, RSNCODE, and RETINFO fields.

To code: Specify the RS-type address, or address in register (2)-(12), of a 100-character field.

,ECB=*ecb*
,ECB=NO_ECB

An optional input parameter field that SDM will post for an asynchronous request. This field is ignored for synchronous requests. The default is NO_ECB.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,WAITTIME=*waittime*
,WAITTIME=0

An optional input parameter field containing how long in seconds SDM will wait for a request to complete. A value of zero says to not time the request. If the time expires before the request is complete a value of RQST_WAITTIME_EXPIRED (see ANTRQSTL) will be returned. If the request is a synchronous request the value will be placed in the RTNCODE field. If the request is an asynchronous request the value will be used as the ECB post code. The default is 0.

To code: Specify the RS-type address, or address in register (2)-(12), of a halfword field.

Subparameters for REQUEST=SSNAP

REQUEST=SSNAP

is used to snap (quickly copy) functional tracks from one RVA device to another.

,SDEVN=*sdevn*
,SVOLSER=*svolser*
,SUCBPTR=*sucbptr*

One of the following three input parameters is required:

,SDEVN=*sdevn*
A parameter field containing the binary device number of the source device you want to snap.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-byte field.

,SVOLSER=*svolser*
A parameter field containing the volser of the source device you want to snap.

To code: Specify the RS-type address, or address in register (2)-(12), of a 6-character field.

,SUCBPTR=*sucbptr*
A parameter field containing the UCB address for the source device you want to snap.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,SEXTINFO=*sextinfo*
A required input parameter field that is an array of eight byte source extent pairs to snap. The first 4-bytes of a pair gives the starting track address and the last 4-bytes gives the ending track address (for a single track the two addresses would be the same).

To code: Specify the RS-type address, or address in register (2)-(12), of a character field.

,SEXTNBR=*sextnbr*
A required input parameter field that tells how many source extent pairs have been supplied in SEXTINFO.

To code: Specify the RS-type address, or address in register (2)-(12), of a halfword field.

,TDEVN=*tdevn*
,TVOLSER=*tvolser*
,TUCBPTR=*tucbptr*
One of the following three input parameters is required:

,TDEVN=*tdevn*
A parameter field containing the binary device number of the target device you want to snap.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-character field.

,TVOLSER=*tvolser*
A parameter field containing the volser of the target device you want to snap.

To code: Specify the RS-type address, or address in register (2)-(12), of a 6-character field.

,TUCBPTR=*tucbptr*
A parameter field containing the UCB address for the target device you want to snap.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,TEXTINFO=*textinfo*
A required input parameter field that is an array of eight byte target extent pairs to snap. The first 4-bytes of a pair gives the starting track address and the last 4-bytes gives the ending track address (for a single track the two addresses would be the same).

To code: Specify the RS-type address, or address in register (2)-(12), of a character field.

,TEXTNBR=*textnbr*
A required input parameter field that tells how many target extent pairs have been supplied in TEXTINFO.

To code: Specify the RS-type address, or address in register (2)-(12), of a halfword field.

,RETINFO=*retinfo*
A required output parameter field that is used to return detailed information about the results of *executing* the request. Information about the results of *scheduling* the request are returned in the RETCODE and RSNCODE fields. Align the area on a word boundary. The first 4-bytes contain the return code and the second 4-bytes contain the reason code. The remainder of the information is dependent on the return and reason codes. See the coding example "SDM API usability guide" on page 575 about how to coordinate the RETCODE, RSNCODE, and RETINFO fields.

To code: Specify the RS-type address, or address in register (2)-(12), of a 100-character field.

,TRELOCATE=*trelocate*
,TRELOCATE=NO
An optional input parameter field containing a keyword that indicates whether to perform cylinder and track relocation of the target extent's count field identifiers. The value is left-justified and padded on the right with blanks. If the field contains binary zeros, the program uses the default.

NO indicates that relocation is not to take place.

YES indicates that relocation is to take place. For example, if a source extent is cylinder 03 track 05, and the corresponding target extent is cylinder 08 track 06, then when the track is copied the count field will be changed on the target device to have a count value of cylinder 08 track 06.

The default is NO.

To code: Specify the RS-type address, or address in register (2)-(12), of a 3-character field.

,ECB=*ecb*
,ECB=NO_ECB
An optional input parameter field that SDM will post for an asynchronous request. This field is ignored for synchronous requests. The default is NO_ECB.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,WAITTIME=*waittime*
,WAITTIME=0
An optional input parameter field containing how long in seconds SDM will wait for a request to complete. A value of zero says to not time the

request. If the time expires before the request is complete a value of RQST_WAITTIME_EXPIRED (see ANTRQSTL) will be returned. If the request is a synchronous request the value will be placed in the RTNCODE field. If the request is an asynchronous request the value will be used as the ECB post code. The default is 0.

To code: Specify the RS-type address, or address in register (2)-(12), of a halfword field.

Character filtering

The rules for a filter pattern are as follows:

- An alphanumeric character represents itself. *Characters are case-sensitive*.
- A question mark (?) represents any character.
- An asterisk (*) represents zero or more characters. Two or more consecutive asterisks are redundant.
- Characters in square brackets ([and]) represent *one of* the enclosed characters. If the first character in brackets is a logical not (shift 6 on the keyboard), the meaning changes to *not one of* the enclosed characters.
- A back-slash (\) allows one of the above characters, or itself, to be used in a pattern.

Parameter descriptions for ILK=XRC

When you invoke the ANTRQST macro for extended remote copy, you can use the following REQUEST parameters. For a list of the Common End Codes associated with each request within this ILK, see “Common end codes for REQUESTS within each ILK” on page 735.

Subparameters for REQUEST=LEVEL

REQUEST=LEVEL
Asks for the level of ANTRQST that is installed on the system.

,RETINFO=*retinfo*
is the name (RS-type) (or address in register (2)-(12) ASM only) of a required 100-character output field used to return the level of ANTRQST installed on the system. If ANTRQST is not installed, return code 7000 (RQST_PC_NUMBER_ZERO - see ANTRQSTL) is returned in register 15 and placed in the RETCODE area, if specified. If ANTRQST is installed, a level value is placed in the first 4-bytes of the RETINFO field. This level value indicates which ANTRQST functions, keywords and parameters are supported. For more information about the LEVEL request output, see the prolog comments in the ANTRQSTL macro (shipped with the ANTRQST macro).

Subparameters for REQUEST=XADD

REQUEST=XADD
Adds pairs of volumes to an XRC session.

SID=*sid*
A required input character field containing the XRC session ID to which the request is sent. The ID is left-justified and padded on the right with blanks.

To code: Specify the RS-type address, or address in register (2)-(12), of an 8-character field.

,PVOLSER=*pvolser*

Belongs to a set of mutually exclusive keys.

To code: Specify the RS-type address, or address in register (2)-(12), of a 6-character field that contains a primary *volser* to be added.

,VOLLIST=*xvollist*

Belongs to a set of mutually exclusive keys. It is the name (RS-type), or address in register (2)-(12), of a variable character input field containing from 2 to 100 volsers of volume pairs to be added. Must be an even number. This field is structured as follows:
- A 1-byte field with a hexadecimal count of the number of volsers in the list.
- A list of 6-byte volsers.

If this parameter is coded parameter SVOLSER is ignored.

,SVOLSER=*svolser*

To code: Specify the RS-type address, or address in register (2)-(12), of a required 6-character field that contains a secondary *volser* to be added.

,COPY=*copy*
,COPY=FUL

An optional input parameter field containing a keyword that indicates the type of volume initialization the system should use to copy the primary volume to the secondary volume. The value is left-justified and padded on the right with blanks. If the field contains binary zeros, the program uses the default.

NO specifies that an initial copy not take place.

FUL specifies that a full initial copy take place.

The default is FUL.

QIK specifies that a quick initial copy take place.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 3-character field.

,ERRLVL=*errlvl*
,ERRLVL=SYSTEM

An optional input parameter field containing an error level that is associated with the volume pair. The value is left-justified and padded on the right with blanks. If the field contains binary zeros, the system program uses the default, which is entered with the XSTART command.

The default is SYSTEM.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 8-character field.

,DONOTBLOCK=*xdonotblock*
,DONOTBLOCK=NO

Part of a mutually exclusive set with DVCBLOCK, DONOTBLOCK is the name (RS-type) (or address in register (2)-(12) ASM only) of an optional 3 character input field containing a keyword that indicates if device blocking should be disallowed. The value is left justified and padded on the right with blanks. If the field contains binary zeros the default will be taken.

Workload-based write pacing affects the behavior of DONOTBLOCK. For more information, refer to "Workload-based write pacing" on page 161.

NO specifies that device blocking is allowed. This is the default.

YES specifies that the volume will be exempted from device blocking and write pacing.

,DVCBLOCK=*xdvcblock*
,DVCBLOCK=ON | WP*n* **| OFF**

Part of a mutually exclusive set with DONOTBLOCK, DVCBLOCK is the name (RS-type) (or address in register (2)-(12) ASM only) of an optional 3-character input field containing a keyword that indicates if device blocking should be enabled or disabled.

Workload-based write pacing affects the behavior of DVCBLOCK. For more information, refer to "Workload-based write pacing" on page 161.

ON specifies that device blocking is enabled.

When workload-based write pacing is in use, XRC automatically converts DVCBLOCK=ON to DVCBLOCK=WP*n*, where *n* is the discretionary level, 6.

WP*n* Specifies that write pacing is to be activated for the specified volume(s), with *n* specifying the level of write pacing delay, 0-F. WP0 specifies that the session default level will be used, as specified the SHADOW DfltWritePacingLvl PARMLIB value. WP1-WP7 results in pacing maximums ranging from 0.02 to 2 ms per recordset, and are useful for volumes with high rates of small blocksize writes, such as data base logs, where minimal response time impact is essential. WP8-WPB results in pacing maximums ranging from 5 to 50 ms per recordset, useful for volumes with high mb/sec write rates. WPC-WPF results in pacing maximums ranging from 100 to 1000 ms per recordset, and should be used in exceptional situations where a high degree of pacing is required.

OFF specifies that both device blocking and write pacing are disabled. The value is left justified and padded on the right with blanks. If the field contains binary zeros the default will be taken.

If neither DVCBLOCK nor DONOTBLOCK is specified, DONOTBLOCK=NO is the default.

,SCSESSION=*scsession*
,SCSESSION=– –

An optional input parameter field containing an alphabetic value that allows specifying a storage control session name for a volume pair. The value is left-justified and padded on the right with blanks. If the field contains binary zeros, the program uses the default.

The default is – –.

SCSESSION cannot be specified when LOGPLUS=YES is specified.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 2-character field.

,LOGPLUS=*xlogplus*
,LOGPLUS=NO | YES

Specifies whether the primary volume of the volume pair is to be explicitly written to by the z/OS System Logger.

YES the primary volume of the volume pair is to be explicitly written to by the z/OS System Logger. When using LOGPLUS, you can specify a single pair or a single pair and a utility pair, but the utility pair must be specified last. A unique storage control session number will be assigned to the primary volume. LOGPLUS=YES cannot be specified with SCSESSION.

NO indicates that LOGPLUS is not used. The default is NO.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 3-character field.

,SUSPENDED=*suspended*

Belongs to a set of mutually exclusive keys. A required input parameter field containing a keyword that indicates if all suspended volume pairs are to be added back into the session. The value is left-justified and padded on the right with blanks. If the field contains binary zeros, the program uses the default.

YES specifies that suspended volume pairs be re-added.

NO specifies that suspended pairs not be re-added.

The default is NO.

To code: Specify the RS-type address, or address in register (2)-(12), of a 3-character field.

,RETINFO=*retinfo*

A required output parameter field that is used to return detailed information about the results of *executing* the request. Information about the results of *scheduling* the request are returned in the RETCODE and RSNCODE fields. Align the area on a word boundary. The first 4-bytes contain the return code and the second 4-bytes contain the reason code. The remainder of the information is dependent on the return and reason codes. See the coding example "SDM API usability guide" on page 575 about how to coordinate the RETCODE, RSNCODE, and RETINFO fields.

To code: Specify the RS-type address, or address in register (2)-(12), of a 100-character field.

,MESSAGES=*messages*

A required output parameter field in which ANTRQST will place the address of a buffer containing messages about the results of executing the request. These are the messages that would have been returned to the caller if the TSO interface had been used. If there are no messages, the address will be zero.

An ANTRQST message buffer has the following format:

- Bytes 1-4 contain the size of the ANTRQST message buffer.
- Byte 5 contains the subpool of the ANTRQST message buffer.
- Byte 6 is not used.
- Bytes 7-8 contain the number of messages in the array that follows.

The rest of the message buffer is an array of messages. Each message has the following format:

- Bytes 1-2 contain the size of the message in bytes 3-x.
- Bytes 3-x contain the message.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 4-byte field.

,ECB=*ecb*
,ECB=NO_ECB
An optional input parameter field that SDM posts for an asynchronous request. This field is ignored for synchronous requests. The default is NO_ECB.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,WAITTIME=*waittime*
,WAITTIME=0
An optional input parameter field containing how long in seconds SDM must wait for a request to complete. A value of zero says to not time the request. If the time expires before the request is complete a value of 7039 (RQST_WAITTIME_EXPIRED [see ANTRQSTL]) is returned. If the request is a synchronous request the value is placed in the RTNCODE part of the RETINFO field. If the request is an asynchronous request the value is used as the ECB post code. The default is 0.

To code: Specify the RS-type address, or address in register (2)-(12), of a halfword field.

Subparameters for REQUEST=XADVANCE

REQUEST=XADVANCE
Recovers an XRC session.

SID=*sid*
A required input character field containing the XRC session ID to which the request is sent. The ID is left-justified and padded on the right with blanks.

To code: Specify the RS-type address, or address in register (2)-(12), of an 8-character field.

,RETINFO=*retinfo*
A required output parameter field that is used to return detailed information about the results of *executing* the request. Information about the results of *scheduling* the request are returned in the RETCODE and RSNCODE fields. Align the area on a word boundary. The first 4-bytes contain the return code and the second 4-bytes contain the reason code. The remainder of the information is dependent on the return and reason codes. See the coding example “SDM API usability guide” on page 575 about how to coordinate the RETCODE, RSNCODE, and RETINFO fields.

To code: Specify the RS-type address, or address in register (2)-(12), of a 100-character field.

,MESSAGES=*messages*
A required output parameter field in which ANTRQST will place the address of a buffer containing messages about the results of executing the request. These are the messages that would have been returned to the caller if the TSO interface had been used. If there are no messages, the address will be zero.

An ANTRQST message buffer has the following format:

- Bytes 1-4 contain the size of the ANTRQST message buffer.
- Byte 5 contains the subpool of the ANTRQST message buffer.
- Byte 6 is not used.
- Bytes 7-8 contain the number of messages in the array that follows.

The rest of the message buffer is an array of messages. Each message has the following format:

- Bytes 1-2 contain the size of the message in bytes 3-x.
- Bytes 3-x contain the message.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 4-byte field.

,ECB=*ecb*
,ECB=NO_ECB
An optional input parameter field that SDM posts for an asynchronous request. This field is ignored for synchronous requests. The default is NO_ECB.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,ETYPE=*etype*
,ETYPE=WAIT
The name of an optional CHAR(16) input field that specifies whether or not ANTRQST will return to the caller before recovery has completely finished. The value is left justified and padded on the right with blanks. If the field contains binary zeros the default will be taken.

WAIT indicates ANTRQST will not return until recovery processing has finished. WAIT is the default.

NOWAIT indicates ANTRQST will return as soon as the recovery processing has been scheduled.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,HLQ=*hlq*
,HLQ=SYS1
An optional input parameter field that contains the high level qualifier that was used on the XSTART command. The qualifier name is left justified and padded on the right with blanks. If the field contains binary zeros, the default value (SYS1) is used.

To code: Specify the RS-type address, or address in register (2)-(12), of an 8–character input field.

,MHLQ=*mhlq*
,MHLQ=SYS1
An optional input parameter field that contains the high level qualifier that was used on the XCOUPLE command for the master session control data set. The qualifier name is left justified and padded on the right with blanks. If the field contains binary zeros, the default value (SYS1) is used.

To code: Specify the RS-type address, or address in register (2)-(12), of an 8–character input field.

,ONLINE=*online*
,ONLINE=NO
An optional input parameter field that contains a keyword indicating what type of test to use to determine if a secondary volume can be used as a recovery volume. The value is left justified and padded on the right with blanks. If the field contains binary zeros, the default value (NO) is used.

NO indicates that only the volume added as a secondary volume can be used as a recovery volume. This volume must be synchronous.

YES indicates that any volume that is online and has a *volser* matching that of a desired secondary volume can be used as a recovery volume.

To code: Specify the RS-type address, or address in register (2)-(12), of a 3–character input field.

,WAITTIME=*waittime*
,WAITTIME=0
An optional input parameter field containing how long in seconds SDM must wait for a request to complete. A value of zero says to not time the request. If the time expires before the request is complete a value of 7039 (RQST_WAITTIME_EXPIRED [see ANTRQSTL]) is returned. If the request is a synchronous request the value is placed in the RTNCODE part of the RETINFO field. If the request is an asynchronous request the value is used as the ECB post code. The default is 0.

To code: Specify the RS-type address, or address in register (2)-(12), of a halfword field.

Subparameters for REQUEST=XCONTIME

REQUEST=XCONTIME
Asks for the current XRC consistency time.

,LOGPLUS=*xlogplus*
,LOGPLUS=NO | YES
An optional input parameter field specifying whether the values for SID and HLQ should be obtained from the DefaultSessionId and DefaultHlq parmlib values. The value is left-justified and padded on the right with blanks. If the field contains binary zeros the default NO is assumed.

NO indicates that the SID and HLQ values are specified in this request.

YES indicates that LOGPLUS is to be used and the values for SID and HLQ are to be obtained from the DefaultSessionId and DefaultHlq parmlib values. LOGPLUS=YES cannot be specified with any other XCONTIME request parameters.

Note: For XRC coupled sessions, issue the XRECOVER request for the session before you issue this request. This ensures that the consistency time returned is the master recoverable time.

To code: Specify the RS-type address, or address in register (2)-(12), of an 3-character field.

,SID=*sid*
A required input parameter field containing the XRC session ID to which the request is sent. The ID is left-justified and padded on the right with blanks. SID is not required when LOGPLUS=YES is specified.

To code: Specify the RS-type address, or address in register (2)-(12), of an 3-character field.

,RETINFO=*retinfo*
A required output parameter field that is used to return detailed information about the results of *executing* the request. Information about the results of *scheduling* the request are returned in the RETCODE and RSNCODE fields. Align the area on a word boundary. The first 4-bytes contain the return code and the second 4-bytes contain the reason code. The remainder of the information is dependent on the return and reason

codes. See the coding example “SDM API usability guide” on page 575 about how to coordinate the RETCODE, RSNCODE, and RETINFO fields.

To code: Specify the RS-type address, or address in register (2)-(12), of a 100-character field.

,TIME=*time*

A required output parameter field used to return the consistency time for the specified XRC session. The returned time is a 64-bit time-of-day (TOD) value in UTC (universal time, coordinated) format. The consistency time returned is the time to which volume pairs in the XRC session were last consistent. The time is taken from the application system clock.

The optional ALET parameter can be used to specify the location of the TIME field.

To code: Specify the RS-type address, or address in register (2)-(12), of an 8-character field.

,ALET=*alet*
,ALET=0

An optional input parameter field that specifies the ALET value to be used if the TIME field has an ALET value different from the one associated with the caller. The default is 0.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,ECB=*ecb*
,ECB=NO_ECB

An optional input parameter field that SDM posts for an asynchronous request. This field is ignored for synchronous requests. The default is NO_ECB.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,HLQ=*hlq*
,HLQ=SYS1

An optional input parameter field containing the high level qualifier that was used on the XSTART command. The name is left-justified and padded on the right with blanks. If the field contains binary zeros, the program uses the default.

The HLQ name will only be used if XRC is not active. In this case the consistency time will be obtained from the XRC state data set that has the supplied high level qualifier in the data set name, and a code of 7113 (RQST_XCONTIME_SAVED_CONSISTENCY_TIME [see ANTRQSTL]) will be returned. The default is SYS1.

To code: Specify the RS-type address, or address in register (2)-(12), of an 8-character field.

,WAITTIME=*waittime*
,WAITTIME=0

An optional input parameter field containing how long in seconds SDM will wait for a request to complete. A value of zero says to not time the request. If the time expires before the request is complete a value of 7039 (RQST_WAITTIME_EXPIRED [see ANTRQSTL]) will be returned. If the request is a synchronous request the value will be placed in the RTNCODE field. If the request is an asynchronous request the value will be used as the ECB post code. The default is 0.

To code: Specify the RS-type address, or address in register (2)-(12), of a halfword field.

Subparameters for REQUEST=XCOUPLE

REQUEST=XCOUPLE
Use to manage connections between a local XRC session and a master XRC session.

,SID=*sid*
A required input parameter field containing the XRC local session ID for the request. The ID is left-justified and padded on the right with blanks.

To code: Specify the RS-type address, or address in register (2)-(12), of an 8-character field.

,ETYPE=*etype*
A required output parameter field that specifies the type of coupling operation to be processed. The value is left justified and padded on the right with blanks. The following four options are available for use:

- ADD — says to add the logical session to the master session.
- DELETE — says to delete the logical session from the master session.
- PURGE — says to purge (cleanup) the logical session from the master data set.
- RELEASE — says to release a HOLD on the master session.

To code: Specify the RS-type address, or address in register (2)-(12), of a 16-character field.

,RETINFO=*retinfo*
A required output parameter field that is used to return detailed information about the results of *executing* the request. Information about the results of *scheduling* the request are returned in the RETCODE and RSNCODE fields. Align the area on a word boundary. The first 4-bytes contain the return code and the second 4-bytes contain the reason code. The remainder of the information is dependent on the return and reason codes. See the coding example “SDM API usability guide” on page 575 about how to coordinate the RETCODE, RSNCODE, and RETINFO fields.

To code: Specify the RS-type address, or address in register (2)-(12), of a 100-character field.

,MESSAGES=*messages*
A required output parameter field in which ANTRQST will place the address of a buffer containing messages about the results of executing the request. These are the messages that would have been returned to the caller if the TSO interface had been used. If there are no messages, the address will be zero.

An ANTRQST message buffer has the following format:

- Bytes 1-4 contain the size of the ANTRQST message buffer.
- Byte 5 contains the subpool of the ANTRQST message buffer.
- Byte 6 is not used.
- Bytes 7-8 contain the number of messages in the array that follows.

The rest of the message buffer is an array of messages. Each message has the following format:

- Bytes 1-2 contain the size of the message in bytes 3-x.

- Bytes 3-x contain the message.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 4-character field.

,ECB=*ecb*
,ECB=NO_ECB

An optional input parameter field that SDM posts for an asynchronous request. This field is ignored for synchronous requests.

The default is NO_ECB.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,HLQ=*hlq*
,HLQ=SYS1

An optional input parameter field containing the high level qualifier of the XRC journal data set. The name is left justified and padded on the right with blanks. If the field contains binary zeros, the default is used.

The default is SYS1.

To code: Specify the RS-type address, or address in register (2)-(12), of an 8-character field.

,MHLQ=*mhlq*
,MHLQ=SYS1

An optional input parameter field containing the high level qualifier of the Master Control data set. The name is left justified and padded on the right with blanks. If the field contains binary zeros, the default is used.

The default is SYS1.

To code: Specify the RS-type address, or address in register (2)-(12), of an 8-character field.

,MID=*mid*
,MID=NO_MID

An optional input parameter field containing the XRC master session ID for the request. The ID is left justified and padded on the right with blanks. If the field contains binary zeros, the default is used.

The default is NO_MID.

To code: Specify the RS-type address, or address in register (2)-(12), of an 8-character field.

,WAITTIME=*waittime*
,WAITTIME=0

An optional input parameter field containing how long in seconds SDM will wait for a request to complete. A value of zero says to not time the request. If the time expires before the request is complete a value of 7039 (RQST_WAITTIME_EXPIRED [see ANTRQSTL]) will be returned. If the request is a synchronous request the value will be placed in the return code part of the RETINFO field. If the request is an asynchronous request, the value will be used as the ECB post code. The default is 0.

To code: Specify the RS-type address, or address in register (2)-(12), of a halfword field.

Subparameters for REQUEST=XDEL

Note: Large batches of XDEL requests issued to Duplex volumes may result in significantly increased session delays. To avoid these delays, consider issuing the XDEL commands while all volumes are suspended, or issuing them in small batches across a period of several minutes.

REQUEST=XDEL
Deletes pairs of volumes from an XRC session.

,SID=*sid*
A required input character field containing the XRC session ID to which the request is sent. The ID is left-justified and padded on the right with blanks.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 8-character field.

,PVOLSER=*pvolser*
A required input field containing the primary volser of a volume pair to be deleted, or ALL if all volume pairs are to be deleted.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 6-character field.

,RETINFO=*retinfo*
A required output parameter field that is used to return detailed information about the results of *executing* the request. Information about the results of *scheduling* the request are returned in the RETCODE and RSNCODE fields. Align the area on a word boundary. The first 4-bytes contain the return code and the second 4-bytes contain the reason code. The remainder of the information is dependent on the return and reason codes. See the coding example “SDM API usability guide” on page 575 about how to coordinate the RETCODE, RSNCODE, and RETINFO fields.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 100-character output field.

,MESSAGES=*messages*
A required output parameter field in which ANTRQST will place the address of a buffer containing messages about the results of executing the request. These are the messages that would have been returned to the caller if the TSO interface had been used. If there are no messages, the address will be zero.

An ANTRQST message buffer has the following format:

- Bytes 1-4 contain the size of the ANTRQST message buffer.
- Byte 5 contains the subpool of the ANTRQST message buffer.
- Byte 6 is not used.
- Bytes 7-8 contain the number of messages in the array that follows.

The rest of the message buffer is an array of messages. Each message has the following format:

- Bytes 1-2 contain the size of the message in bytes 3-x.
- Bytes 3-x contain the message.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 4-character field.

,ECB=*ecb*

,ECB=NO_ECB
An optional input parameter field that SDM posts for an asynchronous request. This field is ignored for synchronous requests.

The default is NO_ECB.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,ATTIME=*attime*
,ATTIME=NO_ATTIME
An optional input parameter field that contains the UTC timestamp of when the volumes are to be deleted. The format is YYYY.DDD HH:MM:SS.THMIJU. This field is required if the ETYPE parameter value ATTIME is specified. It is ignored otherwise.

The default is NO_ATTIME.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 24-character input field.

,ETYPE=*etype*
,ETYPE=IMMEDIATE
An optional input parameter field that specifies when the command will be processed. The value is left-justified and padded on the right with blanks. If the field contains binary zeros, the program uses the default. The following options are available for use:

- ATTIME — indicates the XDEL command will be executed when the specified UTC time is reached.
- CANCEL — indicates an outstanding XDEL command will be deleted.
- DRAIN — indicates the XDEL command will be executed when the consistency group represented by the most recent timestamp of all volumes being deleted has been applied.
- IMMEDIATE — indicates the XDEL command will be executed as soon as the current consistency group has been applied.

The default is IMMEDIATE.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 16-character field.

,WAITTIME=*waittime*
,WAITTIME=0
An optional halfword input parameter field containing how long in seconds SDM will wait for a request to complete. A value of zero says to not time the request. If the time expires before the request is complete a value of 7039 (RQST_WAITTIME_EXPIRED [see ANTRQSTL]) will be returned. If the request is a synchronous request, the value will be placed in the return code part of RETINFO. If the request is an asynchronous request the value will be used as the ECB post code.

The default is 0.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional halfword input field.

Subparameters for REQUEST=XEND

REQUEST=XEND
Ends an XRC session.

,SID=*sid*

A required input character field containing the XRC session ID to which the request is sent. The ID is left-justified and padded on the right with blanks.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 8-character field.

,RETINFO=*retinfo*

A required output parameter field that is used to return detailed information about the results of *executing* the request. Information about the results of *scheduling* the request are returned in the RETCODE and RSNCODE fields. Align the area on a word boundary. The first 4-bytes contain the return code and the second 4-bytes contain the reason code. The remainder of the information is dependent on the return and reason codes. See the coding example “SDM API usability guide” on page 575 about how to coordinate the RETCODE, RSNCODE, and RETINFO fields.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 100-character output field.

,MESSAGES=*messages*

A required output parameter field in which ANTRQST will place the address of a buffer containing messages about the results of executing the request. These are the messages that would have been returned to the caller if the TSO interface had been used. If there are no messages, the address will be zero.

An ANTRQST message buffer has the following format:

- Bytes 1-4 contain the size of the ANTRQST message buffer.
- Byte 5 contains the subpool of the ANTRQST message buffer.
- Byte 6 is not used.
- Bytes 7-8 contain the number of messages in the array that follows.

The rest of the message buffer is an array of messages. Each message has the following format:

- Bytes 1-2 contain the size of the message in bytes 3-x.
- Bytes 3-x contain the message.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 4-character field.

,ECB=*ecb*
,ECB=NO_ECB

An optional input parameter field that SDM posts for an asynchronous request. This field is ignored for synchronous requests.

The default is NO_ECB.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,ATTIME=*attime*
,ATTIME=NO_ATTIME

An optional input parameter field that contains the UTC timestamp of when the session is to end. The format is YYYY.DDD HH:MM:SS.THMIJU. This field is required if the ETYPE parameter value ATTIME is specified. It is ignored otherwise.

The default is NO_ATTIME.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 24-character input field.

,ETYPE=*etype*
,ETYPE=IMMEDIATE

An optional input parameter field that specifies when the command will be executed. The value is left-justified and padded on the right with blanks. If the field contains binary zeros, the program uses the default. The following options are available for use:

- ATTIME — indicates the XEND command will be executed when the specified UTC time is reached.
- CANCEL — indicates an outstanding XEND command will be deleted.
- DRAIN — indicates the XEND command will be executed when the consistency group represented by the most recent timestamp of all volumes being deleted has been applied.
- IMMEDIATE — indicates the XEND command will be executed as soon as the current consistency group has been applied.

IMMEDIATE is the default.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 16-character field.

,MHLQ=*mhlq*
,MHLQ=SYS1

An optional input parameter field containing the high level qualifier of the Master Control data set. The name is left justified and padded on the right with blanks. If the field contains binary zeros, the default is used.

The default is SYS1.

To code: Specify the RS-type address, or address in register (2)-(12), of an 8-character field.

,WAITTIME=*waittime*
,WAITTIME=0

An optional halfword input parameter field containing how long in seconds SDM will wait for a request to complete. A value of zero says to not time the request. If the time expires before the request is complete a value of 7039 (RQST_WAITTIME_EXPIRED [see ANTRQSTL]) will be returned. If the request is a synchronous request, the value will be placed in the RTNCODE part of RETINFO. If the request is an asynchronous request, the value will be used as the ECB post code.

The default is 0.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional halfword input field.

Subparameters for REQUEST=XQUERY

REQUEST=XQUERY

Returns status information on an XRC session. The information will be in the MESSAGES buffer.

,SID=*sid*

A required input character field containing the XRC session ID to which the request is sent. The ID is left-justified and padded on the right with blanks.

When an ETYPE parameter value ENVIRONMENTP, ENVIRONMENTH, or ENVIRONMENTF is specified with an SID value that represents an inactive or undefined session, values for the control address space (ANTAS000) are reported. These values are the default environment values plus any overrides from the ANTXIN00 parmlib member.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 8-character field.

,RETINFO=*retinfo*

A required output parameter field that is used to return detailed information about the results of *executing* the request. Information about the results of *scheduling* the request are returned in the RETCODE and RSNCODE fields. Align the area on a word boundary. The first 4-bytes contain the return code and the second 4-bytes contain the reason code. The remainder of the information is dependent on the return and reason codes. See the coding example "SDM API usability guide" on page 575 about how to coordinate the RETCODE, RSNCODE, and RETINFO fields.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 100-character output field.

FEATURES=*features*

Specifies that the requested report display the LIC features available to the storage controls. A required output parameter used together with the STORAGECONTROL keyword, produces an XQUERY STORAGECONTROL_FEATURES report. The report shows identification, capability, and function enablement information about each storage control session.

XFEATURES=*features*

Specifies that the requested report is an XQUERY STORAGECONTROL XFEATURES report, which shows information related to workload-based write pacing.

,MESSAGES=*messages*

A required output parameter field in which ANTRQST will place the address of a buffer containing messages about the results of executing the request. These are the messages that would have been returned to the caller if the TSO interface had been used. If there are no messages, the address will be zero.

An ANTRQST message buffer has the following format:

- Bytes 1-4 contain the size of the ANTRQST message buffer.
- Byte 5 contains the subpool of the ANTRQST message buffer.
- Byte 6 is not used.
- Bytes 7-8 contain the number of messages in the array that follows.

The rest of the message buffer is an array of messages. Each message has the following format:

- Bytes 1-2 contain the size of the message in bytes 3-x.
- Bytes 3-x contain the message.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 4-character field.

,ECB=*ecb*

,ECB=NO_ECB
An optional input parameter field that SDM posts for an asynchronous request. This field is ignored for synchronous requests.

The default is NO_ECB.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,ETYPE=*etype*
,ETYPE=SUMMARY
An optional input parameter field that specifies which query command to execute. The value is left-justified and padded on the right with blanks. If the field contains binary zeros, the program uses the default. The following options are available:

- CONFIGURATION — says an XQUERY CONFIGURATION report is wanted.
- ENVIRONMENTF — says an XQUERY ENVIRONMENT(FLAG) report is wanted.
- ENVIRONMENTH — says an XQUERY ENVIRONMENT(PATCH) report is wanted.
- ENVIRONMENTP — says an XQUERY ENVIRONMENT(PARMLIB) report is wanted.
- DETAIL — says an XQUERY VOLUME_DETAIL report is wanted.
- DETAILS — says an XQUERY STORAGECONTROL DETAIL report is wanted.
- MASTER — says an XQUERY MASTER report is wanted.
- PACE— says an XQUERY PACE report is wanted.
- SET — says an XQUERY SET report is wanted.
- STORAGECONTROL — says an XQUERY STORAGECONTROL report is wanted.
- SUMMARY — says an XQUERY SUMMARY report is wanted . This is the default.
- VOLUME — says an XQUERY VOLUME report is wanted.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 16-character field.

,MHLQ=*mhlq*
,MHLQ=SYS1
An optional input parameter field containing the high level qualifier of the Master Control data set. The name is left justified and padded on the right with blanks. If the field contains binary zeros, the default is used. This field is used only if the MASTER option is specified in the ETYPE keyword.

The default is SYS1.

To code: Specify the RS-type address, or address in register (2)-(12), of an 8-character field.

,SCSESSION=*scsession*
,SCSESSION=—
An optional input field that contains an alphabetic value that allows requesting that only volumes in the specified storage control session be displayed. The value is left justified and padded on the right with blanks.

If the field contains binary zeros, the default is used. This field is ignored for all ETYPEs except DETAIL and VOLUME.

The default is —.

To code: Specify the RS-type address, or address in register (2)-(12) of an optional 2-character field.

,SSID=*ssid*
An optional input field that contains a 2–byte hexadecimal number, left justified, that allows requesting that only volumes in the specified storage subsystem be displayed. This field is ignored for all ETYPEs except DETAIL and VOLUME.

To code: Specify the RS-type address, or address in register (2)-(12) of an optional 3-character field.

,WAITTIME=*waittime*
,WAITTIME=0
An optional halfword input parameter field containing how long in seconds SDM will wait for a request to complete. A value of zero says to not time the request. If the time expires before the request is complete a value of 7039 (RQST_WAITTIME_EXPIRED [see ANTRQSTL]) will be returned. If the request is a synchronous request, the value will be placed in the return code part of RETINFO. If the request is an asynchronous request, the value will be used as the ECB post code.

The default is 0.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional halfword input field.

Subparameters for REQUEST=XRECOVER

REQUEST=XRECOVER
Recovers an XRC session.

,SID=*sid*
A required input character field containing the XRC session ID to which the request is sent. The ID is left-justified and padded on the right with blanks.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 8-character field.

,RETINFO=*retinfo*
A required output parameter field that is used to return detailed information about the results of *executing* the request. Information about the results of *scheduling* the request are returned in the RETCODE and RSNCODE fields. Align the area on a word boundary. The first 4-bytes contain the return code and the second 4-bytes contain the reason code. The remainder of the information is dependent on the return and reason codes. See the coding example “SDM API usability guide” on page 575 about how to coordinate the RETCODE, RSNCODE, and RETINFO fields.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 100-character output field.

,MESSAGES=*messages*
A required output parameter field in which ANTRQST will place the address of a buffer containing messages about the results of executing the

request. These are the messages that would have been returned to the caller if the TSO interface had been used. If there are no messages, the address will be zero.

An ANTRQST message buffer has the following format:

- Bytes 1-4 contain the size of the ANTRQST message buffer.
- Byte 5 contains the subpool of the ANTRQST message buffer.
- Byte 6 is not used.
- Bytes 7-8 contain the number of messages in the array that follows.

The rest of the message buffer is an array of messages. Each message has the following format:

- Bytes 1-2 contain the size of the message in bytes 3-x.
- Bytes 3-x contain the message.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 4-character field.

,CHECK=*check*
,CHECK=NO

An optional input field that indicates if **only** XRECOVER enhanced checking should be performed, bypassing other XRECOVER processing. This produces an XRECOVER CHECK report. The value is left justified and padded on the right with blanks. If the field contains binary zeros the default is taken. CHECK=YES is mutually exclusive with FORCE=YES. The following options are available:

- NO — Do not perform XRECOVER enhanced checking. NO is the default.
- YES — Perform only XRECOVER enhanced checking, bypassing other XRECOVER processing. This produces an XRECOVER CHECK report.

To code: Specify the RS-type address, or address in register (2)-(12), of a 3-character field.

,ECB=*ecb*
,ECB=NO_ECB

An optional input parameter field that SDM posts for an asynchronous request. This field is ignored for synchronous requests.

The default is NO_ECB.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,ETYPE=*etype*
,ETYPE=WAIT

The name of an optional CHAR(16) input field that specifies whether or not ANTRQST will return to the caller before recovery has completely finished. The value is left justified and padded on the right with blanks. If the field contains binary zeros the default will be taken.

WAIT indicates ANTRQST will not return until recovery processing has finished. WAIT is the default.

NOWAIT indicates ANTRQST will return as soon as the recovery processing has been scheduled.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,FORCE=*foce*
,FORCE=NO

An optional input field that indicates whether or not to perform XRECOVER enhanced checking. The value is left justified and padded on the right with blanks. If the field contains binary zeros the default is taken. FORCE=YES is mutually exclusive with CHECK=YES. The following options are available:

- NO — indicates that enhanced checking should be performed. NO is the default.
- YES — indicates that enhanced checking should not be performed.

To code: Specify the RS-type address, or address in register (2)-(12), of a 3-character field.

,HLQ=*hlq*
,HLQ=SYS1

An optional input parameter field containing the high level qualifier that was used on the XSTART command. The name is left-justified and padded on the right with blanks. If the field contains binary zeros, the program uses the default.

The default is SYS1.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 8-character field.

,MHLQ=*mhlq*
,MHLQ=SYS1

An optional input parameter field that contains the high level qualifier that was used on the XCOUPLE command for the master session control data set. The qualifier name is left justified and padded on the right with blanks. If the field contains binary zeros, the default value (SYS1) is used.

To code: Specify the RS-type address, or address in register (2)-(12), of an 8–character input field.

,ONLINE=*online*
,ONLINE=NO

An optional input parameter field containing a keyword that indicates what type of test will be used to determine if a secondary volume can be used as a recovery volume. The value is left-justified and padded on the right with blanks. If the field contains binary zeros, the program uses the default. The following options are available:

- NO — indicates that only the volume added as a secondary volume can be used as a recovery volume. NO is the default value.
- YES — indicates that any volume that is online and has a volser matching that of a desired secondary volume can be used as a recovery volume. Specifying YES for ONLINE is mutually exclusive with specifying YES for TERTIARY.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 3-character field.

,TERTIARY=*tertiary*
,TERTIARY=NO

An optional input parameter field containing a keyword that indicates what type of test will be used to determine if a secondary volume can be used as a recovery volume. The value is left-justified and padded on the right with blanks. If the field contains binary zeros, the program uses the default. The following options are available:

- NO — indicates that only the volume added as a secondary volume can be used as a recovery volume. NO is the default value.
- YES — indicates that any volume that is online and has a volser matching that of a desired secondary volume can be used as a recovery volume, as long as that volume is not the same secondary volume that was in use when the XRC was suspended or ended. Specifying YES for TERTIARY is mutually exclusive with specifying YES for ONLINE.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 3-character field.

,WAITTIME=*waittime*
,WAITTIME=0

An optional halfword input parameter field containing how long in seconds SDM will wait for a request to complete. A value of zero says to not time the request. If the time expires before the request is complete a value of 7039 (RQST_WAITTIME_EXPIRED [see ANTRQSTL]) will be returned. If the request is a synchronous request, the value will be placed in the return code part of RETINFO. If the request is an asynchronous request, the value will be used as the ECB post code.

The default is 0.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional halfword input field.

Subparameters for REQUEST=XSCSTATS

REQUEST=XSCSTATS

Provides statistics for storage control System Data Mover sessions known to the system this command is executed on.

,QRYSIZE=*qrysize*

A required input field that contains the length of the QRYINFO area.

To code: Specify the RS-type address, or address in register (2)-(12), of a required halfword input field.

,QRYINFO=*qryinfo*

A required variable character output field that is used to return the results of the query.

If the field is not big enough to contain the complete result a return code of 7661 (RQST_XSCSTATS_QRYSIZE_TOO_SMALL) will be placed in the return code part of RETINFO and the total amount of space required will be placed in the reason code for RETINFO.

If the field is big enough to contain the complete result, a return code of 7662 (RQST_XSCSTATS_QRYSIZE_BIG_ENOUGH) will be placed in the return code part of RETINFO, and the total amount of space actually used will be placed in the reason code part of RETINFO.

An ANTRQST XSCSTATS RESULTS buffer has the following format:

- Bytes 1-6 contain the eye-catcher XSTATS.
- Bytes 7-8 contain the number of storage control data mover session entries provided in the remainder of the results buffer. Statistics for both concurrent copy (CC) and extended remote copy (XRC) are provided in the results buffer.

The rest of the results buffer depends on the ETYPE value used on the request. If ETYPE(SUMMARY) was specified or no ETYPE value was specified (the default is SUMMARY), the remainder of the results buffer are 32-byte entries of statistics for each storage control session (as indicated by the number found in bytes 7 and 8). If ETYPE(SINGLE) was specified, a single 32-byte entry is provided followed by 8-byte entries for each of the devices in the single storage control session requested.

Each storage control session entry contains the following 32 bytes of statistics:

- Bytes 1–2 contain the Storage Subsystem Identifier (SSID).
- Bytes 3–4 contain the storage control session identifier (scsession):
  - Byte 3:
    - Bit 1–0
    - Bit 2–0
    - Bit 3–0
    - Bit 4:
      - 0 – Concurrent Copy (CC) session
      - 1 – Extended Remote Copy (XRC) session
    - Bit 5 – request for status came in on the same path group that owns the session
    - Bit 6 – internally terminated
    - Bit 7 – suspended
    - Bit 8 – quiesced
  - Byte 4 is the actual scession number.
- Bytes 5–6 contains the record set count for the scession.
- Byte 7 contains the Channel Connection Address (CCA) for the next utility device to use.
- Byte 8 contains the scession status information:
  - Bit 1 – scession is suspended
  - Bit 2 – in long busy because the sidefile is 60 percent of cache
  - Bit 3 – in long busy because residual count is > 63K
  - Bit 4 – utility device is fixed
  - Bits 5–8 – 0
- Bytes 9–10 contain the size of the largest record set
- Bytes 11–14 contain the number of 4K segments in the sidefile
- Bytes 15–18 contain the number of 4K segments which represents 60 percent of the cache size
- Bytes 19–26 contain the scession's current session time in TOD format.
- Bytes 27–28 contain the time remaining until the scession times out.
- Bytes 29–30 contain the number of devices belonging to the SCESSION.
- Bytes 31–32 contain the number of devices that are in device blocking.
- Byte 33 If ETYPE(SINGLE) is specified, starting at byte 33, statistics for each of the devices in the single control session requested are provided. Bytes 29–30 indicate the number of 8-byte areas that follow. Each 8-byte device statistics area contains the following information:
  - Bytes 1–2 contain the device number of the device. The device number is only available if the device is online to the system which processed the request.

- Bytes 3–4 contain the CCA of the device.
- Bytes 5–6 contain the threshold count for the device. The threshold count is only available for the XRC storage control sessions.
- Bytes 7–8 contain the record set count for the device. The record set count is only available for XRC storage control sessions.

Note:

1. A storage control must have at least one device online for it's statistics to be gathered.
2. A storage control session must have at least one device online for it's complete statistics to be gathered.

To code: Specify the RS-type address, or address in register (2)-(12), of a required variable character output field.

,RETINFO=*retinfo*

A required output parameter field that is used to return detailed information about the results of *executing* the request. Information about the results of *scheduling* the request are returned in the RETCODE and RSNCODE fields. Align the area on a word boundary. The first 4-bytes contain the return code and the second 4-bytes contain the reason code. The remainder of the information is dependent on the return and reason codes. See the coding example “SDM API usability guide” on page 575 about how to coordinate the RETCODE, RSNCODE, and RETINFO fields.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 100-character output field.

,ECB=*ecb*
,ECB=NO_ECB

An optional input parameter field that SDM posts for an asynchronous request. This field is ignored for synchronous requests.

The default is NO_ECB.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,ETYPE=*etype*
,ETYPE=SUMMARY

An optional input parameter field that specifies what type of statistics should be returned. The value is left justified and padded on the right with blanks. The following options are available for use:

- SUMMARY — says to return a summary of the data mover sessions (scsessions) being maintained in the storage controls attached to the system. The format of the returned data is that described by the returned message buffer description WITHOUT the array of device statistics. The field containing the number of devices will be filled in, along with the field telling how many devices are in device blocking. The size of the statistics for each session is 32–bytes, plus 8–bytes for a header. So, if there are five scsessions the size of a buffer to hold all of the information is 168–bytes (8 + 5 * 32).
- SINGLE — says to return a complete set of statistics for a specified scsession. SSID and SCSESSION must be specified to allow locating the session. The format of the returned data is that described by the returned message buffer description. The count found in bytes 7–8 will be one. The size of the returned message buffer is 8–bytes for the header, plus 32–bytes for the general scession statistics, plus 8–bytes for each

device in the scsession. So, if an scsession has five devices the size of the buffer will be 80–bytes (8 + 32 + 5 * 8). An scsession has between 1 and 256 devices.

The default is SUMMARY.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 16-character input field.

,SCSESSION=*scsession*
,SCSESSION=——
An optional character input field that allows specifying what storage control session identifier (SCID on the XQUERY VOLUME_DETAIL_REPORT) will be used to gather statistics. Specify a two byte hexadecimal number, left justified. For example, the storage control session identifier of X'03' would be specified as X'0300'.

The default is —

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 2-character input field.

,SSID=*ssid*
An optional character input field that allows specifying what storage control will be used to gather statistics. A two byte hexadecimal number, left justified.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 3-character input field.

,WAITTIME=*waittime*
,WAITTIME=0
An optional halfword input parameter field containing how long in seconds SDM will wait for a request to complete. A value of zero says to not time the request. If the time expires before the request is complete a value of 7039 (RQST_WAITTIME_EXPIRED [see ANTRQSTL]) will be returned. If the request is a synchronous request, the value will be placed in the return code part of RETINFO. If the request is an asynchronous request, the value will be used as the ECB post code.

The default is 0.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional halfword input field.

Subparameters for REQUEST=XSET

REQUEST=XSET
Changes tuning values for an XRC session. The range of valid values for each parameter are described in the publications that describe the XRC TSO commands.

,SID=*sid*
A required input character field containing the XRC session ID to which the request is sent. The ID is left-justified and padded on the right with blanks.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 8-character field.

,RETINFO=*retinfo*
A required output parameter field that is used to return detailed information about the results of *executing* the request. Information about

the results of *scheduling* the request are returned in the RETCODE and RSNCODE fields. Align the area on a word boundary. The first 4-bytes contain the return code and the second 4-bytes contain the reason code. The remainder of the information is dependent on the return and reason codes. See the coding example "SDM API usability guide" on page 575 about how to coordinate the RETCODE, RSNCODE, and RETINFO fields.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 100-character output field.

,DVCBLOCK=*xdvcblock*
,DVCBLOCK=ON | WP*n* **| OFF**

The name (RS-type) (or address in register (2)-(12) ASM only) of an optional three-character input field containing a keyword that indicates whether device blocking should be enabled or disabled.

Workload-based write pacing affects the behavior of DVCBLOCK. For more information, refer to "Workload-based write pacing" on page 161.

ON specifies that device blocking is enabled.

When workload-based write pacing is in use, XRC automatically converts DVCBLOCK=ON to DVCBLOCK=WP*n*, where *n* is the discretionary level, 6.

WP*n* Specifies that write pacing is to be activated for the specified volume(s), with *n* specifying the level of write pacing delay, 0-F. WP0 specifies that the session default level will be used, as specified the SHADOW DfltWritePacingLvl PARMLIB value. WP1-WP7 results in pacing maximums ranging from 0.02 to 2 ms per recordset, and are useful for volumes with high rates of small blocksize writes, such as data base logs, where minimal response time impact is essential. WP8-WPB results in pacing maximums ranging from 5 to 50 ms per recordset, useful for volumes with high mb/sec write rates. WPC-WPF results in pacing maximums ranging from 100 to 1000 ms per recordset, and should be used in exceptional situations where a high degree of pacing is required.

OFF specifies that device blocking and write pacing are disabled. The value is left justified and padded on the right with blanks. If the field contains binary zeros the default will be taken.

,PVOLSER=*pvolser*
,PVOLSER=ALLB | ALL | ALLDP | ALLP

When you specify DVCBLOCK, an optional six-character input field, with the default of ALLB, specifying the name (RS-type) (or address in register (2)-(12) ASM only) of the primary *volser* of a volume that is to have its device blocking or write pacing status changed. Use ALL if all primary volumes are to have their device blocking status changed.

ALLB specifies that all primary volumes set to DVCBLOCK=ON are to be changed. This is the default.

ALLDP specifies that all primary volumes specified as WP0 are to be changed.

ALLP specifies that all primary volumes specified as WP1 - WPF are to be changed.

,REFRESHP=NO | YES

An optional three-character input field, with the value YES specifying that the system data mover is to examine the capabilities of the specified primary storage control(s) and update the data mover's internal control information accordingly. This allows detection of new capabilities, such as write pacing, without having to suspend or reissue the xadd request for the volume pair. NO is the default.

Specify SSID with REFRESHP to identify the primary storage control(s) to be examined. Specify SSID(ALL) or omit SSID to request that all primary storage controls associated with the session be examined.

REFRESHP cannot be used with TIMEOUT.

,MESSAGES=*messages*

A required output parameter field in which ANTRQST will place the address of a buffer containing messages about the results of executing the request. These are the messages that would have been returned to the caller if the TSO interface had been used. If there are no messages, the address will be zero.

An ANTRQST message buffer has the following format:

- Bytes 1-4 contain the size of the ANTRQST message buffer.
- Byte 5 contains the subpool of the ANTRQST message buffer.
- Byte 6 is not used.
- Bytes 7-8 contain the number of messages in the array that follows.

The rest of the message buffer is an array of messages. Each message has the following format:

- Bytes 1-2 contain the size of the message in bytes 3-x.
- Bytes 3-x contain the message.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 4-character field.

,ECB=*ecb*
,ECB=NO_ECB

An optional input parameter field that SDM posts for an asynchronous request. This field is ignored for synchronous requests.

The default is NO_ECB.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,COPY=*copy*
,COPY=FUL

An optional input field that contains a keyword that indicates what type of volume initialization should be used to copy a primary volume to a secondary volume. The value is left justified and padded on the right with blanks. The following options are available for use:

- FUL — specifies that a full initial copy will take place.
- QIK — specifies that a quick initial copy will take place.

The default is FUL.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 3-character input field.

,PAGEFIX=*pagefix*

,PAGEFIX=8 MB
An optional halfword input parameter field that specifies the number of megabytes assigned to the data mover as permanently page-fixed real storage.

The default is 8 MB.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional halfword input field.

,PMEMBER=*pmember*
,PMEMBER=BINARY_ZEROS
An optional character input field that contains a parmlib member name. The value is left justified and padded on the right with blanks. If the field contains binary zeros, it will be ignored. The following options are available:

- PACTION — an optional 1-character input field that contains a keyword that indicates if the parmlib member is to be verified or verified and then applied.
  - V — indicates that the parmlib member is to be verified. V is the default.
  - A — indicates that the parmlib member is to be verified, and if there are no errors it will be applied.
- PDSNAME — an optional 44-character input field containing the name of a partitioned data set that PMEMBER is found in. The value is left justified and padded on the right with blanks. If the field contains binary zeros, it will be ignored. BINARY_ZEROS is the default.

The default is BINARY_ZEROS.

To code: To code PMEMBER, specify the RS-type address, or address in register (2)-(12), of an optional 8-character input field.

,PRIORITY=*priority*
,PRIORITY=LOAD
An optional character parameter field containing a keyword that indicates how the next volume to be synchronized or resynchronized is selected. The value is left-justified and padded on the right with blanks. If the field contains binary zeros, the program uses the default.

FIFO: says select the next volume based on the order of the XADDPAIR commands.

LOAD: says select the next volume based on which controller has the least load on it.

The default is LOAD.

To code: Specify the RS-type address, or address in register (2)-(12), of a 4-character input field.

,RFREQUENCY=*rfrequency*
,RFREQUENCY=00.00.00
An optional character parameter field that specifies how often the software bit map is toggled. Format is HH.MM.SS. To deactivate time controlled toggling, specify 00.00.00

The default is 00.00.00.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 8-character input field.

,RTRACKS=*rtracks*
,RTRACKS=7500

An optional fullword parameter field that specifies how many tracks have changed before the software bit map is toggled. To deactivate track controlled toggling, specify 0.

The default is 7500.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional fullword input field.

,SCSYNCHP=*scsynchp*
,SCSYNCHP=2

An optional halfword input parameter field that specifies the maximum number of concurrent synchronization or resynchronization tasks allowed per primary storage controller.

The default is 2.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional halfword input field.

,SCSYNCHS=*scsynchs*

A required input parameter field that specifies the maximum number of concurrent synchronization or resynchronization tasks allowed per secondary storage controller.

To code: Specify the RS-type address, or address in register (2)-(12), of a required halfword input field.

,SSID=*ssid*
,SSID=ALL

An optional half-word input field that allows specification of the storage controls that to be processed by the XSET TIMEOUT or REFRESHP command. A value of ALL indicates that all storage controls are to be processed. A two byte, left justified hexadecimal number specifies an individual primary storage control that will be processed.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional halfword input field.

SUSLBUSY=*syslbusy*
,SUSLBUSY=NO | YES

Used to enable or disable the Suspend on Long Busy function. Acceptable values are YES (enable) and NO (disable), with NO as the default.

SUSLBUSY is mutually exclusive with TIMEOUT and REFRESHP.If SSID is also specified, the attributes of the storagecontrol sessions associated with the specified SSID's areimmediately modified. If SSID is not specified, the globalvalue for the session is modified. The global value is used for any new storage control session that is subsequently added.

If the storage control microcode supports the function, enabling will cause the microcode to automatically suspendthe storage control session instead of raising extended longbusy when sidefile limits are exceeded.

If the storage control microcode does not support the function, enabling will cause the data mover to suspend a storage control session as soon as it detects extended long busy due to sidefile exceeding limits. Storage control sessions that do not support suspension will be terminated

Disabling allows the data mover to tolerate the long busy condition for 80% of the storage control session timeout interval, after which mirroring is suspended.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional halfword input field.

,SYNCH=*synch*
,SYNCH=2

An optional halfword input parameter field that specifies the maximum number of concurrent synchronization or resynchronization tasks allowed per session.

The default is 2.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional halfword input field.

,TIMEOUT=*timeout*
,TIMEOUT=USER_DEFINED

An optional character input field that specifies the primary storage control timeout value. The format is HH.MM.SS. The storage control default is usually 00.05.00. Specifying a value of 00.00.00 sets the timeout to 00.05.00 (five minutes).

The default is USER_DEFINED.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 8-character input field.

, UTILITY=*utility*
,UTILITY=FLT

An optional character input field that contains a keyword that indicates what type of utility device support should be enabled when an ADDPAIR is done with a secondary volume of XRCULT. The value is left justified and padded on the right with blanks. The following options are available:

- FIX — specifies that fixed utility device support should be used.
- FLT — specifies that floating utility device support should be used.

The default is FLT (FLOAT).

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 3-character input field.

,WAITTIME=*waittime*
,WAITTIME=0

An optional halfword input parameter field containing how long in seconds SDM will wait for a request to complete. A value of zero says to not time the request. If the time expires before the request is complete a value of 7039 (RQST_WAITTIME_EXPIRED [see ANTRQSTL]) will be returned. If the request is a synchronous request the value will be placed in the RTNCODE part of RETINFO. If the request is an asynchronous request the value will be used as the ECB post code.

The default is 0.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional halfword input field.

Subparameters for REQUEST=XSTART

REQUEST=XSTART
Starts up an XRC session.

,SID=*sid*
A required input character field containing the XRC session ID to which the request is sent. The ID is left-justified and padded on the right with blanks.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 8-character field.

,STYPE=*stype*
A required input character field containing the type of XRC session identified by SID. The id is left-justified and padded on the right with blanks.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 8-character field.

,ERRLVL=*errlvl*
A required character input character field containing an error level that is associated with volume pairs added to the session. The value is left-justified and padded on the right with blanks.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 8-character field.

,RETINFO=*retinfo*
A required output parameter field that is used to return detailed information about the results of *executing* the request. Information about the results of *scheduling* the request are returned in the RETCODE and RSNCODE fields. Align the area on a word boundary. The first 4-bytes contain the return code and the second 4-bytes contain the reason code. The remainder of the information is dependent on the return and reason codes. See the coding example "SDM API usability guide" on page 575 about how to coordinate the RETCODE, RSNCODE, and RETINFO fields.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 100-character output field.

,MESSAGES=*messages*
A required output parameter field in which ANTRQST will place the address of a buffer containing messages about the results of executing the request. These are the messages that would have been returned to the caller if the TSO interface had been used. If there are no messages, the address will be zero.

An ANTRQST message buffer has the following format:

- Bytes 1-4 contain the size of the ANTRQST message buffer.
- Byte 5 contains the subpool of the ANTRQST message buffer.
- Byte 6 is not used.
- Bytes 7-8 contain the number of messages in the array that follows.

The rest of the message buffer is an array of messages. Each message has the following format:

- Bytes 1-2 contain the size of the message in bytes 3-x.
- Bytes 3-x contain the message.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 4-character field.

,ECB=*ecb*
,ECB=NO_ECB
An optional input parameter field that SDM posts for an asynchronous request. This field is ignored for synchronous requests.

The default is NO_ECB.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,HLQ=*hlq*
,HLQ=SYS1
An optional input parameter field containing the high level qualifier of the state, control, and journal data sets. The name is left-justified and padded on the right with blanks. If the field contains binary zeros, the program uses the default.

The default is SYS1.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 8-character field.

,WAITTIME=*waittime*
,WAITTIME=0
An optional halfword input parameter field containing how long in seconds SDM will wait for a request to complete. A value of zero says to not time the request. If the time expires before the request is complete a value of 7039 (RQST_WAITTIME_EXPIRED [see ANTRQSTL]) will be returned. If the request is a synchronous request the value will be placed in the RTNCODE part of RETINFO. If the request is an asynchronous request the value will be used as the ECB post code.

The default is 0.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional halfword input field.

,MODE=*mode-list*
Specifies the mode for XRC. This is intended for IBM use only.

Subparameters for REQUEST=XSTATUS

REQUEST=XSTATUS
Returns status information on an XRC session. The information will be in the MESSAGES buffer.

,RETINFO=*retinfo*
A required output parameter field that is used to return detailed information about the results of *executing* the request. Information about the results of *scheduling* the request are returned in the RETCODE and RSNCODE fields. Align the area on a word boundary. The first 4-bytes contain the return code and the second 4-bytes contain the reason code. The remainder of the information is dependent on the return and reason codes. See the coding example “SDM API usability guide” on page 575 about how to coordinate the RETCODE, RSNCODE, and RETINFO fields.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 100-character output field.

,MESSAGES=*messages*
A required output parameter field in which ANTRQST will place the address of a buffer containing messages about the results of executing the request. These are the messages that would have been returned to the caller if the TSO interface had been used. If there are no messages, the address will be zero.

An ANTRQST message buffer has the following format:

- Bytes 1-4 contain the size of the ANTRQST message buffer.
- Byte 5 contains the subpool of the ANTRQST message buffer.
- Byte 6 is not used.
- Bytes 7-8 contain the number of messages in the array that follows.

The rest of the message buffer is an array of messages. Each message has the following format:

- Bytes 1-2 contain the size of the message in bytes 3-x.
- Bytes 3-x contain the message.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 4-character field.

,ECB=*ecb*
,ECB=NO_ECB
An optional input parameter field that SDM posts for an asynchronous request. This field is ignored for synchronous requests.

The default is NO_ECB.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,ETYPE=*etype*
,ETYPE=ADDRSPACE
Specifies that XRC display the names of all active XRC address spaces in the LPAR, including the session name, the address space name and identifier, the type of session, the associated cluster name, and associated master session name.

,ETYPE=*etype*
,ETYPE=CLUSTER
Specifies that SDM display information about coupled XRC sessions associated with the cluster session in the LPAR where the command is processed, including the cluster session name, XRC session name, session status, volume status, interlock status, journal delta time, RCV/ADV delta time, master session name and HLQ, maximum journal delta for all sessions in the cluster session, and maximum RCV/ADV delta time for all sessions in the cluster session.

,MHLQ=*mhlq*
,MHLQ=SYS1
An optional input parameter field containing the high level qualifier of the Master Control data set. The name is left justified and padded on the right with blanks. If the field contains binary zeros, the default is used. This field is used only if the MASTER option is specified in the ETYPE keyword.

The default is SYS1.

To code: Specify the RS-type address, or address in register (2)-(12), of an 8-character field.

,WAITTIME=*waittime*
,WAITTIME=0

An optional halfword input parameter field containing how long in seconds SDM will wait for a request to complete. A value of zero says to not time the request. If the time expires before the request is complete a value of 7039 (RQST_WAITTIME_EXPIRED [see ANTRQSTL]) will be returned. If the request is a synchronous request, the value will be placed in the return code part of RETINFO. If the request is an asynchronous request, the value will be used as the ECB post code.

The default is 0.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional halfword input field.

DATASET

Specifies that XRC direct the XSTATUS output to the sequential data set that is identified by *dataset_name.* Standard TSO naming conventions apply for *dataset_name.* The target data set must be a sequential data set and must be large enough to accommodate the XSTATUS reports you want to collect. You can specify DATASET with or without DISP; the default is DISP(OLD).

Allocate a data set for the XSTATUS function with the following attributes:

```
DCB=(RECFM=FB,LRECL=80,BLKSIZE=6400,DSORG=PS)
```

DISP

Specifies how XRC writes the XSTATUS output to the data set, as follows:

MOD Specifies that XRC append the output to the data set.

OLD Specifies that XRC clear the data set before it receives output.

SHR Specifies that XRC clear the data set before it receives output. SHR also allows multiple allocations of the same data set.

If you specify DISP, you must also specify DATASET.

MSGROUTEID

Specifies the user ID to which XRC messages that are associated with the processing of this command are routed. If the specified user ID is not logged on, TSO saves messages in the TSO BRODCAST data set and displays them after the next logon.

Note: Messages issued by the initial command parser routine are sent to the user ID that issues the TSO command. If you specify both DATASET and MSGROUTEID, XQUERY output is directed to the data set that is specified with DATASET and not to the MSGROUTEID user ID. Error messages are directed to the MSGROUTEID user ID.

Subparameters for REQUEST=XSUSPEND

REQUEST=XSUSPEND

Suspends an XRC session, or suspends volume pairs in an XRC session.

,SID=*sid*

A required input character field containing the XRC session ID to which the request is sent. The ID is left-justified and padded on the right with blanks.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 8-character field.

The following is a set of mutually exclusive keys. This set is required; **only one key must** be specified.

,PVOLSER=*pvolser*

Belongs to a set of mutually exclusive keys. A character parameter field that contains the primary *volser* of a volume pair to be suspended, or ALL if all volume pairs are to be suspended.

To code: Specify the RS-type address, or address in register (2)-(12), of a 6-character field.

,TIMEOUT=*timeout*

Belongs to a set of mutually exclusive keys. A character parameter field containing a timeout value that will be sent to every controller in the session as part of the session suspension. The format is HH.MM.SS. To get the storage control default specify 00.00.00.

To code: Specify the RS-type address, or address in register (2)-(12), of an 8-character field.

,SCESSION=*scsession*

Belongs to a set of mutually exclusive keys. A character input field containing a hexadecimal storage control session ID, left justified, of a session to be suspended. For example, scession X'0A' would be specified as X'0A00'.

In addition, you must use **DEVN** with SCSESSION. DEVN is a required 2-character input field that contains a hexadecimal number of a device attached to the storage control that will have it's session suspended. For example, device number F40 would be specified as X'0F40'.

To code: Specify the RS-type address, or address in register (2)-(12), of a 2-character field for both SCSESSION and DEVN.

End of a set of mutually exclusive required keys.

,RETINFO=*retinfo*

A required output parameter field that is used to return detailed information about the results of *executing* the request. Information about the results of *scheduling* the request are returned in the RETCODE and RSNCODE fields. Align the area on a word boundary. The first 4-bytes contain the return code and the second 4-bytes contain the reason code. The remainder of the information is dependent on the return and reason codes. See the coding example “SDM API usability guide” on page 575 about how to coordinate the RETCODE, RSNCODE, and RETINFO fields.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 100-character output field.

,MESSAGES=*messages*

A required output parameter field in which ANTRQST will place the address of a buffer containing messages about the results of executing the request. These are the messages that would have been returned to the caller if the TSO interface had been used. If there are no messages, the address will be zero.

An ANTRQST message buffer has the following format:

- Bytes 1-4 contain the size of the ANTRQST message buffer.
- Byte 5 contains the subpool of the ANTRQST message buffer.
- Byte 6 is not used.
- Bytes 7-8 contain the number of messages in the array that follows.

The rest of the message buffer is an array of messages. Each message has the following format:

- Bytes 1-2 contain the size of the message in bytes 3-x.
- Bytes 3-x contain the message.

To code: Specify the RS-type address, or address in register (2)-(12), of a required 4-character field.

,ECB=*ecb*
,ECB=NO_ECB

An optional input parameter field that SDM posts for an asynchronous request. This field is ignored for synchronous requests.

The default is NO_ECB.

To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

,ATTIME=*attime*
,ATTIME=NO_ATTIME

An optional input parameter field that contains the UTC timestamp of when the suspend will happen. The format is YYYY.DDD HH:MM:SS.THMIJU. This field is required if the ETYPE parameter value ATTIME is specified. It is ignored otherwise.

The default is NO_ATTIME.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 24-character input field.

,ETYPE=*etype*
,ETYPE=IMMEDIATE

An optional input parameter field that specifies when the command will be executed. The value is left-justified and padded on the right with blanks. If the field contains binary zeros, the program uses the default.

ATTIME indicates the XSUSPEND command will be executed when the specified UTC time is reached.

CANCEL indicates an outstanding XSUSPEND command will be deleted.

DRAIN indicates the XSUSPEND command will be executed when the consistency group represented by the most recent timestamp of all volumes being deleted has been applied.

IMMEDIATE indicates the XSUSPEND command will be executed as soon as the current consistency group has been applied.

The default is IMMEDIATE.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional 16-character field.

,MHLQ=*mhlq*
,MHLQ=SYS1

An optional input parameter field containing the high level qualifier of the Master Control data set. The name is left justified and padded on the right with blanks. If the field contains binary zeros, the default is used.

The default is SYS1.

To code: Specify the RS-type address, or address in register (2)-(12), of an 8-character field.

,WAITTIME=*waittime*
,WAITTIME=0

An optional halfword input parameter field containing how long in seconds SDM will wait for a request to complete. A value of zero says to not time the request. If the time expires before the request is complete a value of 7039 (RQST_WAITTIME_EXPIRED [see ANTRQSTL]) will be returned. If the request is a synchronous request the value will be placed in the RTNCODE part of RETINFO. If the request is an asynchronous request the value will be used as the ECB post code.

The default is 0.

To code: Specify the RS-type address, or address in register (2)-(12), of an optional halfword input field.

Common end codes for REQUESTS within each ILK

End Codes for Requests

The following list of codes is used at the end of each request (for example, REQUEST=XADD, REQUEST=SDVCINFO) within an ILK. Because the values stay the same for each request , there is no need to display the descriptions more than once.

,RETCODE=*retcode*

An optional output parameter into which the return code is to be copied from GPR 15.

To code: Specify the RS-type address of a fullword field, or register (2)-(12).

,RSNCODE=*rsncode*

An optional output parameter into which the reason code is to be copied from GPR 0.

To code: Specify the RS-type address of a fullword field, or register (2)-(12).

,PLISTVER=IMPLIED_VERSION
,PLISTVER=MAX
,PLISTVER=0

An optional input parameter that specifies the version of the macro. PLISTVER determines which parameter list the system generates. PLISTVER is an optional input parameter on all forms of the macro, including the list form. When using PLISTVER, specify it on all macro forms used for a request and with the same value on all of the macro forms. The values are as follows:

- **IMPLIED_VERSION**, which is the lowest version that allows all parameters specified on the request to be processed. If you omit the PLISTVER parameter, IMPLIED_VERSION is the default.
- **MAX**, if you want the parameter list to be the largest size currently possible. This size might grow from release to release and affect the amount of storage that your program needs.

  If you can tolerate the size change, IBM recommends that you always specify PLISTVER=MAX on the list form of the macro. Specifying MAX ensures that the list-form parameter list is always long enough to hold all the parameters you might specify on the execute form, when both are assembled with the same level of the system. In this way, MAX ensures that the parameter list does not overwrite nearby storage.
- **0**, if you use the currently available parameters.

To code: Specify one of the following:

- IMPLIED_VERSION

- MAX
- A decimal value of 0

,MF=S
,MF=(L,*list addr***)**
,MF=(L,*list addr,attr***)**
,MF=(L,*list addr***,'X'0D)**
,MF=(E,*list addr***)**
,MF=(E,*list addr***,COMPLETE)**
,MF=(E,*list addr***,NOCHECK)**
,MF=(M,*list addr***)**
,MF=(M,*list addr***,COMPLETE)**
,MF=(M,*list addr***, NOCHECK)**

An optional input parameter that specifies the macro form.

Use MF=S to specify the standard form of the macro, which builds an inline parameter list and generates the macro invocation to transfer control to the service. MF=S is the default.

Use MF=L to specify the list form of the macro. Use the list form together with the execute form of the macro for applications that require reentrant code. The list form defines an area of storage that the execute form uses to store the parameters. Only the PLISTVER parameter may be coded with the list form of the macro.

Use MF=E to specify the execute form of the macro. Use the execute form together with the list form of the macro for applications that require reentrant code. The execute form of the macro stores the parameters into the storage area defined by the list form, and generates the macro invocation to transfer control to the service.

Use MF=M to add or remove a keyword to an existing ANTRQST parameter list. If the keyword references a parameter area with user data, the pointer to the parameter is added to or removed from the specified ANTRQST parameter list. If the keyword references a parameter that can only have certain values, the ANTRQST parameter list is updated with the specified parameter value. The ANTRQST macro parameter list has parameters by reference as well as parameters by value.

If a keyword is required for the MF=E form of ANTRQST, but one is not specified on the initial MF=L form, use of the MF=M form to add the required keyword will not copy the keyword parameter value into the ANTRQST parameter list.

Use of the MF=M to add or remove keywords and parameters requires extra instructions at program execution time. All keywords can be specified on the initial MF=L setup of the ANTRQST parameter list. The program can then set the parameter field to a known value when needed, and set the parameter field to binary zero when the keyword is not needed. Having the application program set the parameter field to binary zero reduces the extra instructions needed to support the MF=M form of adding or removing a keyword.

IBM recommends that you use the modify and execute forms of ANTRQST in the following order:

- Use ANTRQST ...MF=(M,list-addr,COMPLETE) specifying appropriate parameters, including all required ones.
- Use ANTRQST ...MF=(M,list-addr,NOCHECK), specifying the parameters that you want to change.
- Use ANTRQST ...MF=(E,list-addr,NOCHECK), to execute the macro.

,*list addr*
The name of a storage area to contain the parameters. For MF=S, MF=E, and MF=M, this can be an RS-type address or an address in register (1)-(12).

,*attr*
An optional 1 to 60-character input string that you use to force boundary alignment of the parameter list. Use a value of X'0F' to force the parameter list to a word boundary, or X'0D' to force the parameter list to a doubleword boundary. If you do not code *attr*, the system provides a value of X'0D'.

,COMPLETE
Specifies that the system is to check for required parameters and supply defaults for omitted optional parameters.

,NOCHECK
Specifies that the system is not to check for required parameters and is not to supply defaults for omitted optional parameters.

ANTQFRVL macro

This macro maps the input and output fields in the parameter area pointed to by the VOLLIST keyword in the ANTRQST Query Fast Replication Volumes request.

The ANTRQST REQUEST=QFRVOLS VOLLIST parameter has the name of an area mapped by the ANTQFRVL macro. This area is required for the QFRVOLS request of ANTRQST.

The header area must be filled in by the caller. This includes setting QFRVLEYE to `ANTQFRVL`, setting QFRVLVRL to the value of the QFRVRLC symbol and setting QFRVLLEN to the total length, in bytes, of the parameter area.

The caller sets the number of volume entries, supplies the volume or device information for them and sets all other volume entry areas to binary zero.

ANTQFRVL header return and reason codes

When ANTRQST returns to the caller, and the RETINFO return and reason code are zero, the ANTQFRVL header return code and reason code have the following values:

Return Code	Reason Code	Description
0	0 (X'00')	The control volume and VOLLIST have been processed. All volumes are eligible.
0	1 (X'01')	The control volume and VOLLIST volumes have been processed. One or more VOLLIST volumes are ineligible.
4	1 (X'01')	No volumes have been processed. The control volume does not support and version of FlashCopy.
4	2 (X'02')	No volumes have been processed. The control volume could not be accessed to determine FlashCopy support status.
8	x	Input parameters or VOLLIST extent parameters invalid. This is an IBM internal error.
12	x	QFRVOLS abend or other serious error. This is an IBM internal error.

Volume capability and reason codes

When ANTRQST returns to the caller, and the RETINFO return and reason code are zero, and the ANTQFRVL header return code is zero, each volume entry has one of the following values:

Volume Capable	Volume Reason	Description
1	0 (X'00')	The volume is source capable.
2	0 (X'00')	The volume is target capable.
0	.	The volume is not currently Fast Replication capable. The volume reason will have a non-zero value indicating the current cause for being incapable.
0	1 (X'01')	The maximum number of FlashCopy relationships are active on the volume.
0	2 (X'02')	The volume is a PPRC Primary volume.
0	3 (X'03')	The volume is a PPRC Secondary volume
0	4 (X'04')	The volume is a Concurrent Copy source.
0	5 (X'05')	The volume is extended Remote Copy source.
0	6 (X'06')	The volume is currently inhibited from starting any FlashCopy operations.
0	7 (X'07')	A volume level (phase 1) FlashCopy relationship is active on this volume.
0	8 (X'08')	The maximum number of FlashCopy relationships for the Enterprise Storage Server (ESS) are active.
0	9 (X'09')	The volume is currently inaccessible.
0	10 (X'0A')	Asynchronous PPRC Primary target active for volume.
0	11 (X'0B')	Full volume target relation exists on the volume.
0	12 (X'0C')	Full volume source relation exists on the volume.
0	13 (X'0D')	Maximum full volume relations exist on the volume.
0	14 (X'0E')	Volume is space efficient, target capability requested and SETGTOK=YES was not specified, or Preserve Mirror was specified.
0	15 (X'0F')	The volume is in the process of a PPRC cascading failover/failback recovery operation or the PPRC volume was established with incremental resynchronization started, and target capability checking was requested.
0	16 (X'10')	The volume is a cascaded PPRC volume.
0	17 (X'11')	The volume is not capable of data set level FlashCopy operations. (Space efficient volumes are not eligible for data set level FlashCopy operations.)
0	18 (X'12')	The volume is not capable of inhibiting writes to the source.
0	200 (X'C8')	The volume cannot be found.
0	201 (X'C9')	The volume does not support FlashCopy.
0	202 (X'CA')	The volume and the control volume are not in the same subsystem.

Volume Capable	Volume Reason	Description
0	203 (X'CB')	An I/O error was detected when obtaining FlashCopy status of the volume. To obtain a diagnostic ABEND dump from the ANTMAIN address space that will provide additional information about the error, issue system command F ANTAS000,CTFLG ABEND_LIC ON, and resubmit the failing request. Once the dump has been obtained, issue system command F ANTAS000,CTFLG ABEND_LIC OFF.
0	204 (X'CC')	The volume does not have the same track geometry as the control volume.
0	205 (X'CD')	A VM formatted MVS minidisk.
0	206 (X'CE')	An I/O timeout was detected when obtaining FlashCopy status of the volume.
0	207 (X'CF')	The specified extent is outside the volume's capability.

Preserve Mirror reason codes

When ANTRQST returns to the caller, and the RETINFO return and reason code are zero, and the ANTQFRVL header return code is zero, and Version 2 or higher is specified, each volume entry has one of the values described below.

Preserve Mirror Capable	Preserve Mirror Reason	Description
1	0 (X'00')	The volume is Preserve Mirror Capable.
2	0 (X'00')	Preserve Mirror checking not requested.
3	0 (X'00')	Preserve Mirror checking was not done because the volume is not FlashCopy capable.
0	.	The volume is not currently Preserve Mirror capable. The volume reason will have a non-zero value indicating the current cause for being incapable.
0	1 (X'01')	The maximum number of FlashCopy relations are active on the volume.
0	2 (X'02')	Reserved
0	3 (X'03')	Reserved
0	4 (X'04')	The volume's secondary is a Concurrent Copy source.
0	5 (X'05')	The volume's secondary is an XRC primary.
0	6 (X'06')	The volume's secondary currently is inhibited from starting any FlashCopy operations.
0	7 (X'07')	A Volume Level (Phase 1) FlashCopy relation is active on this volume's secondary.
0	8 (X'08')	The maximum number of FlashCopy relations for the volume's secondary volume's SFI.
0	9 (X'09')	The volume's secondary is inaccessible.
0	10 (X'0A')	The volume's secondary is a Global Mirror primary.
0	11 (X'0B')	A full volume target relationship already exists in the volume's secondary.

Preserve Mirror Capable	Preserve Mirror Reason	Description
0	12 (X'0C')	A full volume source relationship already exists on the volume's secondary and target capability was requested.
0	13 (X'0D')	The maximum number of full volume source relationships exist on the volume's secondary and source capability was requested.
0	14 (X'0E')	The volume's secondary is space efficient.
0	15 (X'0F')	The volume's secondary is in the process of a PPRC cascading failover/failback recovery or the PPRC volume was established with incremental resync. Target capability is required.
0	16 (X'10')	The volume's secondary is a cascaded PPRC volume.
0	17 (X'11')	The volume's secondary is not capable of dataset-evel FlashCopy operations,
0	18 (X'12')	The volume's secondary is not capable of inhibiting writes to the source.
0	19-127 (X'13-1B')	Reserved
0	128 (X'80')	The volume is not a PPRC primary device. For target checking, software can treat this as capable.
0	129 (X'81')	The volume's secondary and the control volume's secondary are not in the same SFI.
0	130 (X'82')	The volume's PPRC relationship is not full duplex.
0	131 (X'83')	The volume's secondary's SFI does not have the microcode installed that supports IBM Remote Pair FlashCopy (Preserve Mirror).
0	132 (X'84')	The control volume is not a PPRC primary.
0	133 (X'85')	The command was received while the subsystem was installing a new microcode load and the level of one or both CECs does not support IBM Remote Pair FlashCopy (Preserve Mirror).
0	134 (X'86')	An error was detected when sending the command to the volume's secondary.
0	135 (X'87')	The volume is space efficient and target capability was requested. Space efficient volumes are not allowed as the target of a Preserve Mirror operation.
0	208 (X'D0')	The volume is a PPRC primary and the Preserve Mirror feature is not enabled.
0	209 (X'D1')	PPRC Primary not Full Duplex or Preserve Mirror version 2 not installed.
0	256 (X'100')	The controller does not have the Preserve Mirror feature enabled.

Volume extent capability and reason codes

When ANTRQST returns to the caller, and the RETINFO return and reason code are zero, and the ANTQFRVL header return code is zero, each volume entry has one of the following values in the extent capability and reason fields if the volume is capable and the extents are specified.

Extent Capable	Extent Reason	Description
1	0 (X'00')	The extent is source capable.
2	0 (X'00')	The extent is target capable.
0	.	The extent is not currently Fast Replication capable. The extent reason will have a non-zero value indicating the current cause for being incapable.
0	1 (X'01')	For a target eligibility request, the extent has active FlashCopy target tracks.
0	2 (X'02')	For source or target eligibility requests, the extent has active FlashCopy target tracks.
0	3 (X'03')	For a source eligibility request, the extent has the maximum number of source relationships to target tracks.
0	4 (X'04')	The specified extent is in a validation required status.

Linkage = %INCLUDE SYSLIB(ANTQFRVL)

Input = "%QFRVL_BASE" can be used to change the default basing attribute of the control block. "QFRVL_BASE = 'NONE'" will cause the declare to not be used. "%QFRVL_BASE = 'BASED(R1)'" will cause the declared to based on R1.

Preserve Mirror reason codes

When ANTRQST returns to the caller from an ANTRQST QFRVOLS request that specifies that Preserve Mirror checking is to be done, and the RETINFO return and reason code, as well as the ANTQFRVL header return code, are all zero, then each volume entry has one of the following values. Note that there is a capability check for Preserve Mirror but not a capability check for Preserve Mirror Preferred or Preserve Mirror Required. The software will need to consider the returned information and make a determination if a Preserve Mirror Preferred request may be successful.

Preserve Mirror Capable	Preserve Mirror Capable Reason	Description
1	0 (X'00')	The request is Preserve Mirror capable.
0	-	The request is not Preserve Mirror capable. The reason will have a non-zero value indicating the current cause for being incapable.
0	1 (X'01')	Maximum number of FlashCopy relationships exist for the secondary volume.
0	4 (X'04')	The volume's secondary is a concurrent copy source volume.
0	5 (X'05')	The volume's secondary is an XRC source volume.
0	6 (X'06')	The volume's secondary is currently inhibited from starting any FlashCopy operations.
0	7 (X'07')	A volume level (version 1) FlashCopy relationship is active on the volume's secondary.
0	8 (X'08')	The maximum number of relationships has been reached for the volume's secondary's control unit.

Preserve Mirror Capable	Preserve Mirror Capable Reason	Description
0	9 (X'09')	The volume's secondary is inaccessible; capability information could not be obtained.
0	10 (X'0A')	The volume's secondary is a cascaded Global Mirror primary.
0	11 (X'0B')	A full volume target relationship exists on the volume's secondary.
0	12 (X'0C')	A full volume source relationship exists on the volume's secondary and target capability checking was requested.
0	13 (X'0D')	The maximum number of full volume source relationships exist on the secondary volume and source checking capability checking was requested.
0	128 (X'80')	The volume is not a PPRC primary volume. For target checking, this can be considered capable, as it would be a FlashCopy to a simplex device. DFSMS considers this capable.
0	129 (X'81')	The volume's secondary and the control volume's secondary are not in the same Storage Facility Image (SFI).
0	130 (X'82')	The volume's PPRC relationship is not full duplex.
0	131 (X'83')	The volume's secondary control unit does not have the microcode installed that supports a preserve mirror operation.
0	132 (X'84')	The control volume is not a PPRC primary device.
0	133 (X'85')	The command was received while the subsystem was attempting to install a new microcode load and the microcode level of one or both CECs does not support a Preserve Mirror operation.
0	134 (X'86')	An error was detected when sending the FlashCopy capability command to the volume's secondary.
0	135 (X'87')	The volume is space efficient and target capability was requested.

ABEND codes

None.

Return and reason codes

When the ANTRQST macro returns control to your program the following occurs:

- General purpose register (GPR) 15 (and *retcode*, when you code RETCODE) contains a return code.
- When the value in GPR 15 is not zero, GPR 0 (and *rsncode*, when you code RSNCODE) contains reason code.

There are two ways that return and reason codes can be returned to the caller:

- In the RTNCODE and RSNCODE fields. These codes are returned by the system data mover (SDM) module ANTCDPI that runs in the caller's address space. They are associated with the validation of common parameters supplied on the ANTRQST macro (ASYNCH, for example) along with verifying the environment (for example, a PPRC request that address space ANTAS000 is active). The parameters from the ANTRQST invocation are placed in a parameter list (called the ANTRQSTL).

- In the return code and reason code part of the RETINFO field. Once the request has been transferred to the appropriate SDM address space, there is no mechanism provided by z/OS to get information back to the caller's RTNCODE and RSNCODE fields (actually registers 15 and 0). The codes returned in the RETINFO field have to do with additional validation of parameters, scheduling of processing that runs in an SDM address space, and the results of executing the request. For ILK=XRC requests that also specify a MESSAGES subparameter, the RETINFO return code reflects only the results of scheduling the request to the XRC data mover address space. If scheduling is successful, the execution results are placed in the message buffer specified by the MESSAGES subparameter.

Return codes will be found in either RTNCODE or RETINFO, but not both. EXCEPTION: return code 0 can be found in both places.

Executable macro ANTRQSTL has been provided to include definitions and descriptions of the return code and reason code values. These descriptions can also be found in *z/OS MVS System Messages, Vol 1 (ABA-AOM)*.

For ANTRQST return and reason codes, refer to *z/OS MVS System Messages, Vol 1 (ABA-AOM)*. It identifies hexadecimal and decimal return codes and reason codes. IBM support personnel may request the entire reason code, SYSLOG, and any associated dumps.

Appendix D. REXX support for the ANTRQST API

Programming Interface Information

This section describes using the REXX programming language with the ANTRQST API.

The program ANTTREXX allows a REXX exec to use the ANTRQST API. The function that is available with ANTTREXX is described in Table 66.

Table 66. Functions Supported by ANTTREXX

Function (ILK)	Request Types	Purpose	Refer To
ESSRVCS (FlashCopy, Global Mirror and DIAG)	FCESTABLISH	Establish a FlashCopy relationship	"FCESTABLISH (FlashCopy establish)" on page 753
	FCQUERY	Query a FlashCopy relationship	"FCQUERY (FlashCopy query)" on page 759
	FCWITHDRAW	Remove (withdraw) a FlashCopy relationship	"FCWITHDRAW (FlashCopy withdraw)" on page 762
	LEVEL	Query the level of ANTRQST that is installed on the system	"LEVEL (query ANTRQST level)" on page 767
	QHA	Query a device and determine where path groups are established	"QHA (Query where path groups are established)" on page 767
	RQUERY	Query a Global Mirror session	"RQUERY (Global Mirror query)" on page 787
	RSESSION	Control a Global Mirror session	"RSESSION (control a Global Mirror session)" on page 790
	RVOLUME	Manage volumes for Global Mirror	"RVOLUME (Global Mirror manage volumes)" on page 792
	STATESAVE	Request a state save (standard or non-disruptive)	"STATESAVE (Request State Save)" on page 795

Table 66. Functions Supported by ANTTREXX (continued)

Function (ILK)	Request Types	Purpose	Refer To
PPRC (peer-to-peer remote copy)	FENCE	Prevent unintended access of a device through Soft Fence, or to prevent any hosts from bringing a device online through SPID Fence. Can be applied to a device, a set of devices in a Logical Subsystem (LSS), or all devices in an LSS. This invokes ANTRQST ILK=PPRC REQUEST=FENCE.	"FENCE (PPRC prevent unintended access through Soft Fence)" on page 768
	LEVEL	Query the level of ANTRQST that is installed on the system	"LEVEL (query ANTRQST level)" on page 767
	PDELPAIR	Delete a PPRC volume pair	"PDELPAIR (PPRC delete volume pair)" on page 769
	PDELPATH	Delete a PPRC path	"PDELPATH (PPRC delete path)" on page 770
	PESTPAIR	Establish a PPRC volume pair	"PESTPAIR (PPRC establish volume pair)" on page 772
	PESTPATH	Establish a PPRC path	"PESTPATH (PPRC establish path)" on page 776
	PFREEZE	Perform a PPRC freeze, to control volume groups	"PFREEZE (PPRC control volume groups)" on page 778
	PQSCSTAT	Query storage controller status	"PQSCSTAT (Query storage controller)" on page 779
	PQUERY	Query PPRC status	"PQUERY (PPRC query)" on page 780
	PRECOVER	Recover data on the recovery system	"PRECOVER (PPRC recover data on the recovery system)" on page 782
	PRUN	Perform a PPRC run, to control volume groups	"PRUN (PPRC control volume group)" on page 784
	PSETCHAR	Set characteristics of a PPRC volume pair	"PSETCHAR (PPRC set characteristics of volume pair)" on page 785
PSUSPEND	Suspend a PPRC volume pair	"PSUSPEND (PPRC suspend volume pair)" on page 786	

Table 66. Functions Supported by ANTTREXX (continued)

Function (ILK)	Request Types	Purpose	Refer To
XRC (extended remote copy)	LEVEL	Query the level of ANTRQST that is installed on the system	"LEVEL (query ANTRQST level)" on page 767
	XADD	Add XRC volume pairs	"XADD (XRC add volume pair)" on page 796
	XADVANCE	Recover an XRC session	"XADVANCE (XRC recover)" on page 798
	XCONTIME	Query the current XRC consistency time	"XCONTIME (XRC consistency time)" on page 799
	XCOUPLE	Manage connections between a local XRC session and a master XRC session	"XCOUPLE (couple XRC connections)" on page 799
	XDEL	Delete pairs of volumes from an XRC session	"XDEL (XRC delete volume pair)" on page 800
	XEND	End an XRC session	"XEND (XRC end)" on page 801
	XQUERY	Returns status information on an XRC session	"XQUERY (XRC query)" on page 802
	XRECOVER	Recover an XRC session	"XRECOVER (XRC recover)" on page 803
	XSCSTATS	Provide XRC statistics	"XSCSTATS (XRC statistics)" on page 805
	XSET	Change tuning values for an XRC session	"XSET (XRC session parameters)" on page 806
	XSTART	Start up an XRC session	"XSTART (XRC start)" on page 808
	XSTATUS	Query XRC status	"XSTATUS (XRC status)" on page 809
	XSUSPEND	Suspend an XRC session or suspend volume pairs in an XRC session	"XSUSPEND (XRC suspend)" on page 810
-	WAIT	Perform a wait (invoke STIMER)	"WAIT (perform a wait)" on page 796

With just a few exceptions, the keywords you use with ANTTREXX are the same as the keywords you use with ANTRQST. Some keywords, such as ECB and ASYNCH, are not applicable to the REXX environment, so are not valid for the ANTTREXX program. See Table 67 for details.

Table 67. Keyword Differences between ANTTREXX and ANTQRST

ANTQRST Keyword	For Request Type	ANTTREXX
ALET	FCQUERY, FCWITHDRAW, QFRVOLS, PQUERY, QHA, XCONTIME	Not supported
ASYNCH	Any	Not supported
ECB	Any	Not supported
RETINFO	FCESTABLISH, FCQUERY, FCWITHDRAW, FENCE, QHA	The results are returned, but you do not use the keyword
SRCEXTENTS	FCESTABLISH, FCWITHDRAW	Use SRCEXTN or, to specify a value of ALL, SRCEXTNA()
TGTEXTENTS	FCESTABLISH, FCWITHDRAW	Use TGTEXTN or, to specify a value of ALL, TGTEXTNA()

Table 67. Keyword Differences between ANTTREXX and ANTQRST (continued)

ANTQRST Keyword	For Request Type	ANTTREXX
WAITTIME	FENCE, QHA	Not supported

Calling ANTTREXX

This topic describes how to call ANTTREXX.

Syntax for ANTTREXX

You call ANTTREXX as follows:

```
►►──┬─CALL──ANTTREXX──'──request-type──'──┬──────────────────────────────┬──────────────────┬──────►◄
    │                                     └─,──'──compound-variable──'─┘                  │
    └─result──=──ANTTREXX──(──?──request-type──?──┬─────────────────────────────┬──)─┘
                                                  └─,──?──compound-variable──?─┘
```

where:

request-type
: specifies the request type, such as FCESTABLISH. It is required. For a list of request types, see Table 66 on page 745.

  This corresponds to the ANTRQST REQUEST parameter.

compound-variable
: Is a REXX compound variable that contains keywords and parameters for the request. These vary with the request type. They are described in "ANTTREXX request types" on page 752.

  The keywords and parameters are not case-sensitive.

  The syntax for keywords is *keyword*(*value*), for example, `DEVN(0A00)`. Do not use embedded blanks, for example, `DEVN (0A00)` or `DEVN( 0A00)`.

result
: is the result returned by ANTTREXX. See "ANTTREXX results" for more information.

ANTTREXX results

Error results from ANTTREXX are returned in REXX variable RESULT. It is good programming practice to check the RESULT variable after each ANTTREXX call.

Table 68. Results of Calling ANTTREXX

Error Type	Contents of RESULT
None	Message ANTR8810I for LEVEL or query requests and message ANTR8810I for all other requests
Detected by ANTRQST	Message ANTR8811E
Detected by ANTTREXX. These are usually parameter or syntax errors.	Message describing the error. Any unprintable characters in the results will have been changed to periods. Blanks will have been changed to underscores (_).

For a description of the ANTR8810I and ANTR8811E messages, see *z/OS MVS System Messages, Vol 1 (ABA-AOM)*.

To display the contents of the RESULT variable, you can use the REXX SAY instruction, for example `SAY RESULT`.

Information returned from an ANTTREXX call replaces output fields in the compound variable. For example, the QRYINFO keyword defines an output field for an FCQUERY request. After an ANTTREXX call, you reference the information that is returned by using the variable that is associated with QRYINFO.

Example of specifying keywords

Issue an FCQUERY request with the DEVN, QRYSIZE, FORMAT and QRYINFO keywords.

1. Define the keywords in variable INOUT, which is the stem of a REXX compound variable:
   ```
   INOUT.0 = 4
   INOUT.1 = 'DEVN(0A00)'
   INOUT.2 = 'QRYSIZE(2000)'
   INOUT.3 = 'FORMAT(FQMAP)'
   INOUT.4 = 'QRYINFO()'
   ```
   Note that the value of INOUT.0 is a count of the number of values in the stem.
2. Call ANTTREXX with an FCQUERY request, specifying variable INOUT:
   ```
   CALL ANTTREXX 'FCQUERY' , 'INOUT'
   ```
3. Display the information from the query. The keyword for the output field is QRYINFO, which is in INOUT.4.
   ```
   SAY INOUT.4
   ```

Programming requirements

Because ANTTREXX invokes the ANTRQST API, the authorization and system requirements are the same for ANTTREXX as they are for ANTRQST.

The user identifier found in the ACEEUSRI field of the ACEE is used for all access authorization checking. This user identifier must have read access to a RACF profile in the FACILITY class:

- For FlashCopy and Global Mirror for ESS requests, the profile is STGADMIN.ANT.ESFC.COMMANDS. In addition, FCQUERY and RQUERY requests require profile STGADMIN.ANT.ESFC.FCQUERY.
- For PPRC requests, the profile is STGADMIN.ANT.PPRC.COMMANDS. In addition, PQUERY requests require profile STGADMIN.ANT.PPRC.CQUERY. FENCE requests require profile STGADMIN.ANT.PPRC.FENCE.
- For XRC requests, the profile is STGADMIN.ANT.XRC.COMMANDS. In addition, XQUERY and XSTATUS requests require profile STGADMIN.ANT.XRC.XQUERY.
- For QHA requests the profile is STGADMIN.ANT.ESS.QHA.
- For STATESAVE requests the profile is STGADMIN.ANT.ESS.STATESAVE.

For FlashCopy and Global Mirror requests, the ANTMAIN address space must be active. For XRC and PPRC requests, the ANTAS000 address space must be active.

SDM API does not provide any data management services, such as allocation or cataloging of data sets. To copy data sets where these services are needed (such as accessing data sets on target devices), use a data set copy program such as DFSMSdss.

Samples

DFSMS provides sample REXX programs in SYS1.SAMPLIB and in DGTCLIB. You can invoke them from ISPF option 6 or from TSO batch by invoking IKJEFT01 in your JCL, as follows:

```
//stepname EXEC PGM=IKJEFT01
```

The samples are:

- ANTFREXX
  Invokes FlashCopy commands
- ANTPREXX
  Invokes PPRC commands
- ANTRREXX
  Invokes Global Mirror commands
- ANTXREXX
  Invokes XRC commands

Sample ANTFREXX, which invokes FCESTABLISH, FCWITHDRAW and FCQUERY, is shown in Table 69 on page 751.

Table 69. Sample REXX Exec

Exec | **Description**

```
/*** REXX **************************************************************/
/*
```

← Process input parameters

```
 * Parse the input parameters.  They will be converted to upper-case
 * before any processing is done. The first parameter will be saved
 * in field 'command' and the rest will be saved in 'operands'
 */

parse upper arg command operands

/*
 * Get a count of how many parameters are left.
 */

args.0 = WORDS(operands)

/*
 * Make sure at least one parameter has been entered.
 */

if args.0 = 0 then
do
  say 'Required parameters are missing'
  exit
end
```

← Process the variables and look for the QRYINFO keyword

```
/*
 * Loop through the remaining parameters. They will be saved in stem
 * variable 'args'. Save the index value for QRYINFO if it is entered.
 */

q = 0

do i=1 to args.0
  parse var operands vbl operands
  args.i=vbl
  if vbl = 'QRYINFO()' then
    q = i
end
```

← Call ANTTREXX

```
call anttrexx command , 'args'
```

← Process the RESULT variable

```
presult = result
say result
if resultx <> 'RESULTX' then
  say resultx

parse var presult resulta resultb resultc resultd resulte resultrest
if q <> 0 then
 do
```

Table 69. Sample REXX Exec (continued)

Exec	Description

`if resultd = 'RETCODE' then`
`  if resulte = 7622 then`
`  do`
`    say 'QRYSIZE TOO SMALL, RESULTS TRUNCATED. QRYINFO:'`
`    say args.q`
`  end`
`  else`
`  if resulte = 7623 then`
`    say args.q`
`else`
`if resultb = 'OK' then`
`  say args.q`
`end`
 	
`if datatype(result3.0,'n') then do`
`  say /* blank line */`
`  do i = 1 to result3.0`
`    say result3.i`
`  end`
`end`
 	
`if resultb = 'OK' then`
`  exit 0`
`else`
`  exit 8`
	← Evaluate the return code for QRYSIZE and display appropriate information

ANTTREXX request types

This topic describes the request types that are supported with ANTTREXX:

- "RVOLUME (Global Mirror manage volumes)" on page 792
- "STATESAVE (Request State Save)" on page 795
- "WAIT (perform a wait)" on page 796
- "XADD (XRC add volume pair)" on page 796
- "XADVANCE (XRC recover)" on page 798
- "XCONTIME (XRC consistency time)" on page 799
- "XCOUPLE (couple XRC connections)" on page 799
- "XDEL (XRC delete volume pair)" on page 800
- "XEND (XRC end)" on page 801
- "XQUERY (XRC query)" on page 802
- "XRECOVER (XRC recover)" on page 803
- "XSCSTATS (XRC statistics)" on page 805
- "XSET (XRC session parameters)" on page 806
- "XSTART (XRC start)" on page 808
- "XSTATUS (XRC status)" on page 809
- "XSUSPEND (XRC suspend)" on page 810

FCESTABLISH (FlashCopy establish)

Use request type FCESTABLISH to establish a FlashCopy relationship. This invokes ANTRQST ILK=ESSRVCS REQUEST=FCESTABLISH.

SDEVN(*sdevn***)**
: Specifies the hexadecimal device number of a local CKD device that will be used as the FlashCopy source. SDEVN is mutually exclusive with REMOTE(YES) and OPENDVCS(YES).

TDEVN(*tdevn***)**
: Specifies the hexadecimal device number of a local CKD device that will be used as the FlashCopy target. TDEVN is mutually exclusive with REMOTE(YES) and OPENDVCS(YES).

DEVN(*devn***)**
: Specifies the hexadecimal device number of a local CKD device that will be used as an access for an inband request, with REMOTE(YES), or fixed block request, with OPENDVCS(YES).

If you use OPENDVCS(YES) without REMOTE(YES), DEVN must specify a CKD access volume located in the same subsystem cluster as the fixed block device identified by SOURCE in this command.

If you use OPENDVCS(YES) with REMOTE(YES), DEVN must specify a CKD access volume located in the same subsystem cluster as the PPRC primary device that is paired with the PPRC secondary that is specified as the FlashCopy source.

If you use REMOTE(YES) without OPENDVCS(YES), DEVN must specify the PPRC primary device that is paired with the PPRC secondary that is specified as the FlashCopy source.

SRCSERIAL(*srcserial***)**
: Specifies the 10-digit hexadecimal storage control serial number for the FlashCopy source. SRCSERIAL is required when DEVN is specified.

SRCLSS(*srclss***)**
: Specifies the 2-digit hexadecimal logical subsystem number for the FlashCopy source. SRCLSS is required when DEVN is specified.

SRCDVC(*srcdvc*)
Specifies the 2-digit hexadecimal value for the FlashCopy source device (either connection address or logical unit number). SRCDVC is required when DEVN is specified.

SRCSSID(*srcssid*)
When REMOTE(YES) is specified without OPENDVCS(YES), this specifies the 4-digit hexadecimal value for the subsystem where the FlashCopy is to occur. This must be the same value as that specified for SSID on the PPRC secondary volume on the PPRC establish pair command.

TGTSERIAL(*tgtserial*)
Specifies the 10-digit hexadecimal storage control serial number for the FlashCopy target. TGTSERIAL is required when either DEVN or TGTUCB(NO) is specified.

TGTLSS(*tgtlss*)
Specifies the 2-digit hexadecimal logical subsystem number for the FlashCopy target. TGTLSS is required when either DEVN or TGTUCB(NO) is specified.

TGTDVC(*tgtdvc*)
Specifies the 2-digit hexadecimal value for the FlashCopy target device (either connection address or logical unit number). TGTDVC is required when either DEVN or TGTUCB(NO) is specified.

SRCEXTNA()
Indicates that all tracks of the source device are active in the FlashCopy relationship (this is a full volume request). Either SRCEXTNA or SRCEXTN is required when SDEVN is specified. SRCEXTNA is mutually exclusive with SRCEXTN.

This corresponds to the ANTRQST keyword SRCEXTENTS, with a value of ALL.

SRCEXTN(*srcextent*,*srcextent*,*srcextent*,*srcextent*)
Indicates track address ranges on the source CKD volume that is active in this FlashCopy relationship. Each extent range consists of a starting track address and an ending track address. Each source extent range must have a related target extent range, specified in the TGTEXTN keyword, identifying the same number of tracks. The tracks identified in the source extent fields must be valid for the source device.

Either SRCEXTNA or SRCEXTN is required when SDEVN is specified. SRCEXTN is mutually exclusive with SRCEXTNA.

Supply the value for SRCEXTN in SUPERZAP format, using a comma between every 4 digits. For example, to copy source tracks 01110002 through 011100C, use SRCEXTN(0111,0002,0111,000C). To specify multiple source extent ranges, make each range a separate keyword. For instance to copy source tracks 01110002 through 011100C as well as tracks 02220003 through 0222000D, you would use this:

```
SRCEXTN(0111,0002,0111,000C)
SRCEXTN(0222,0003,0222,000D)
```

instead of this:

```
SRCEXTN(0111,0002,0111,000C,0222,0003,0222,000D)
```

You can supply up to 40 extent ranges on a single FCESTABLSIH request.

This corresponds to the ANTRQST keyword SRCEXTENTS.

TGTEXTNA()

Indicates that this establish is a full volume request. Either TGTEXTNA or TGTEXTN is required when SDEVN is specified. TGTEXTNA is mutually exclusive with TGTEXTN.

This corresponds to the ANTRQST keyword TGTEXTENTS with a value of ALL.

TGTEXTN(*tgtextent***,***tgtextent***,***tgtextent***,***tgtextent***)**

Indicates track address ranges on the target CKD volume that is active in this FlashCopy relationship. Each extent range consists of a starting track address and an ending track address. Each target extent range must have a related source extent range, specified with the SRCEXTN keyword, identifying the same number of tracks. The tracks identified in the target extent fields must be valid for the target device.

Either TGTEXTNA or TGTEXTN is required when TDEVN is specified. TGTEXTN is mutually exclusive with TGTEXTNA

Supply the value for TGTEXTN in SUPERZAP format, with a comma between every 4 digits. For example, to copy to target tracks 01110002 through 011100C, you would use TGTEXTN(0111,0002,0111,000C). To supply multiple source extent ranges, specify each range as a separate keyword. For instance, to copy to target tracks 01110002 through 011100C as well as to tracks 02220003 through 0222000D, you would use this:

```
TGTEXTN(0111,0002,0111,000C)
SRCEXTN(0222,0003,0222,000D)
```

instead of this:

```
TGTEXTN(0111,0002,0111,000C,0222,0003,0222,000D)
```

You can supply up to 40 extent ranges on a single FCESTABLSIH request.

This corresponds to the ANTRQST keyword TGTEXTENTS.

OPENDVCS(NO | YES)

Specifies whether Open System (fixed block) devices or CKD devices are addressed by this command.

NO The required parameters (SDEVN and TDEVN) identify CKD devices. This is the default.

YES

The required parameters (DEVN, SRCSERIAL, SRCLSS, SRCSSID, SRCDVC, TGTSERIAL, TGTLSS, TGTDVC) identify fixed block devices using a CKD access device.

REMOTE(NO | YES)

Indicates whether the request is for a subsystem that is not directly attached to the issuing processor.

NO The request is directed to a device on a subsystem locally attached to the issuing processor. This is the default.

YES

The request is directed to a remote subsystem using inband subsystem functions. When you specify REMOTE(YES), the request must be for a full volume.

With an inband request for a CKD secondary device, the DEVN keyword must identify a PPRC primary device in a subsystem accessible by the host processor issuing the request, and the FlashCopy source device must be the PPRC secondary of that DEVN.

With an inband request for an Open System (fixed block) secondary device, the DEVN keyword must identify an online CKD device located in the same subsystem cluster as the fixed block device of the PPRC primary that is paired with the PPRC secondary that is designated as the FlashCopy source.

INCREMENTAL(NO | YES | YTW)
Specifies whether the FlashCopy establish relationship remains active after the initial copy is complete. If the FlashCopy relationship remains active, subsequent changes are tracked, and future FlashCopy operations require only a subset of the volume to be copied.

NO The FlashCopy relationship ends after the background copy has completed, when MODE(COPY) is specified, or all source and target tracks have been updated, when MODE(NOCOPY) is specified. This is the default.

YES
The FlashCopy relationship remains in effect after the request has completed. Subsequent changes are tracked so that future FlashCopy operations are performed incrementally. This relationship continues until explicitly terminated with a FlashCopy Withdraw request.

YTW
The FlashCopy relationship remains in effect after the request has completed. Subsequent changes are tracked so that future FlashCopy operations are performed incrementally. This relationship continues until it is ended explicitly with a FlashCopy Withdraw request.

Note: The FlashCopy target is writable while the incremental relationship is active. Any writes done to the target during this period are overwritten if a subsequent increment is done, keeping the target a true copy of the source. If the relationship is reversed, the changes made to the target are reflected on the source.

TGTPPRIM(NO | YES)
Specifies whether the target in the FlashCopy relationship can be the primary in a PPRC pair.

NO The target in this FlashCopy relationship cannot be the primary in a PPRC pair. This is the default.

YES
The target in this FlashCopy relationship can be the primary in a PPRC pair. This request proceeds normally to the specified target but the hardware ignores the PPRC status of the target.

MODE(COPY | NOCOPY | NO2CPY | ASYNC)
Specifies the type of FlashCopy relationship to be started for this pair of volumes.

COPY
Establish a FlashCopy relationship between the source device and the target device. The request will start a background copy of all tracks within the specified extents from the source volume to the target volume. When the background copy completes, the FlashCopy relationship terminates. The target volume extents contain the same data as the source volume extents when the FlashCopy relationship was first established.

NOCOPY
: Establish a FlashCopy relationship between the source device and the target device. The program does not do a background copy of tracks from the source volume to the target volume. Processing of data on the source volume is the same as if the source volume were not in a FlashCopy relationship. Records read from the target volume within the specified extents will have the same data as the related source volume records at the time the FlashCopy relationship was established.

When a FlashCopy NOCOPY relationship is ended, the track data on the target device is unpredictable and should not be used. If updates occur to source device tracks in the FlashCopy NOCOPY relationship, a copy of the source tracks from the point in time of the FlashCopy establish may or may not be written to the target device.

Tracks may be copied from the source to the target volume even if the source track is not changed. This includes the track that contains the volume label. Therefore, to avoid duplicate volume serial problems when the target device is later varied online, IBM recommends that you label the target volume again after withdrawing a volume-level FlashCopy NOCOPY relationship.

NO2CPY
: Initiate a background copy from the source device to the target device. When a MODE=NOCOPY relationship already exists between the source and the target, this relationship ends when the background copy is completed. There must be an existing FlashCopy relationship between the source and the target. If one does not exist, none will be created.

ASYNC
: Indicates that this is being established to a volume set that is currently, or will be, part of a Global Mirror session.

ONLINTGT(NO | YES)
: Indicates whether the FlashCopy Establish should continue if the specified target device is in an online state to any system.

NO Fail the FlashCopy Establish if the target device is online to any system. This is the default.

YES
: Continue the FlashCopy Establish if the target device is online to any system.

ACTION(FREEZE | FRR)
: Specifies an action that is to occur during the FlashCopy establish request.

FREEZE
: The FlashCopy source volume is to be part of a FlashCopy consistency group. The FlashCopy relationship is established between the source and target volumes, or extents, and all I/O to the source volume is held, resulting in a long busy, until one of the following conditions is met:

- A FlashCopy withdraw with action THAW is processed by the LSS where the volume resides
- A two-minute timer has expired. The two-minute timer can be adjusted using the ESS Specialist GUI.

FRR
: Fast Reverse Restore. This reverses the direction of the FlashCopy relationship when change recording is active, restoring the source volume

to the state it was in when it last flashed to the target. Changed tracks are copied back from the target to the source. At the completion of Fast Reverse Restore, the original target volume is no longer usable.

If you specify ACTION, the following are ignored:
- INCREMENTAL(YES)
- MODE(ASYNC)
- MODE(NO2CPY)
- SRCEXTN
- TGTEXTN

SETGTOK(NO | YES)
Specifies whether the target of the specified full volume relationship can be a space efficient volume.

NO The target cannot be a space efficient volume. This is the default.

YES
The target can be a space efficient volume. If an out-of-pace condition occurs, the relationship is failed.

MSGREQ(YES | NO)
Specifies whether to wait for FlashCopy Establish initialization to complete.

NO Do not wait. This is the default.

YES
Wait. MSGREQ(YES) is ignored if MODE(NOCOPY) or MODE(NO2CPY) is specified.

TGTUCB(YES | NO)
Specifies whether an MVS device number is used to address the target. The values are:

YES
The SDEVN and TDEVN keywords are used to identify the source and target devices in the relationship. The MVS device numbers are used in the SDEVN and TDEVN keywords to address the source and target devices. This is the default.

NO The MVS device number is specified in the SDEVN keyword to identify the source. The serial number, LSS, and CCA are specified in the TGTSERIAL, TGTLSS, and TGTDVC keywords to address the target device.

TGTUCB(NO) is not applicable when either OPENDVCS(YES) or REMOTE(YES) is specified.

PRESMIR(PREF | REQ | NO)
Indicates the handling of the request based on whether the specified target is a PPRC primary device. PRESMIR(PREF) and PRESMIR(REQ) are mutually exclusive with REMOTE(YES) and ACTION(FRR).

TGTPPRIM(YES) is required with PRESMIR(PREF) or PRESMIR(REQ).

REQ
If the specified target device is a Metro Mirror primary device, the pair must not go into a duplex pending state as the result of this FCESTABLISH request.

PREF
If the specified target device is a Metro Mirror primary device, it would be

preferable that the pair does not go into a duplex pending state as a result of the FCESTABLISH request, but if a Preserve Mirror operation cannot be accomplished, the establish request should still be performed.

NO The FCESTABLISH request is to be performed without considering a Preserve Mirror operation. This is the default.

WAITTIME(*waittime* **|** **0)**
Specifies how long in seconds SDM will wait for a request to complete. A value of zero specifies that the request should not be timed.

If the wait time expires before the request is complete, the program returns a return code of 7039 (RQST_WAITTIME_EXPIRED). For more information about RQST_WAITTIME_EXPIRED, refer to the ANTRQSTL macro.

SUBCHSET(*subchset* **|** **0)**
Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is supported by the host system's processor and configured for the device.

If you omit the parameter, the command is issued to the device that is logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

TSUBCHSET(*tsubchset* **|** **0)**
Specifies which subchannel set is to be used to get information about the target device specified with the TDEVN parameter. This is the subchannel set for the device as defined in the Hardware Configuration Dialog (HCD). The values are determined by what is currently supported by the host system's processor and configured for the device.

If you omit the parameter, the device that is currently logically in subchannel set 0 will be used. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), via HCD.

FCQUERY (FlashCopy query)

Use request type FCQUERY to query a FlashCopy relationship. This invokes ANTRQST ILK=ESSRVCS REQUEST=FCQUERY.

DEVN(*devn***)**
Specifies the hexadecimal device number of a local CKD device that will be used for the query request. DEVN is required.

When querying a local device, this specifies the 4–digit hexadecimal device number of the device being queried.

When querying a local fixed block device (OPENDVCS(YES) is specified), DEVN specifies a CKD access volume located in the same subsystem cluster as the fixed block device identified by the QRYDVC keyword.

When querying a remote fixed block device (both OPENDVCS(YES) and REMOTE(YES) are specified), DEVN specifies a CKD access volume located in the same subsystem cluster as the PPRC primary device that is paired with the PPRC secondary device that is being queried.

When querying a remote CKD device (REMOTE(YES) is specified), DEVN specifies a PPRC primary device that is paired with the PPRC secondary device that is being queried.

QRYINFO()
Creates the data area where the query information is to be returned. This area is the size specified with the QRYSIZE keyword. QRYINFO is required.

QRYSIZE(*qrysize*)
Specifies the decimal length of the QRYINFO area. It is required.

QRYSERIAL(*qryserial*)
Specifies the 10-digit hexadecimal storage control serial number for the device being queried. QRYSERIAL is required with REMOTE(YES) or OPENDVCS(YES).

QRYLSS(*qrylss*)
Specifies the 2-digit hexadecimal logical subsystem number for device being queried. QRYLSS is required with REMOTE(YES) or OPENDVCS(YES).

QRYDVC(*qrydvc*)
Specifies the 2-digit hexadecimal value for the device being queried, either the connection address or the logical unit number. QRYDVC is required with REMOTE(YES) or OPENDVCS(YES).

QRYSSID(*qrydvc*)
When REMOTE(YES) is specified without OPENDVCS(YES), this keyword specifies the 4-digit hexadecimal value for subsystem for the device being queried. This must be the same value as that specified for SSID on the PPRC secondary volume on the PPRC establish pair command.

OPENDVCS(NO | YES)
Specifies whether Open System (fixed block) devices or CKD devices are addressed by this command.

NO
The parameters identify a CKD device. This is the default.

YES
The parameters (QRYSERIAL, QRYLSS, QRYDVC) identify a fixed block device using a CKD access device.

REMOTE(NO | YES)
Specifies whether the request is for a subsystem that is not directly attached to the issuing processor.

NO
The request is directed to a device on a subsystem locally attached to the issuing processor. This is the default.

YES
The request is directed to a remote subsystem using inband subsystem functions.

When specifying an inband request for a CKD secondary device, DEVN must identify a PPRC primary device in a subsystem accessible by the host processor issuing the request, and the device being queried must be the PPRC secondary of that DEVN.

When specifying an inband request for an Open System (fixed block) secondary device, DEVN must identify an online CKD device located in the same subsystem cluster as the fixed block device of the PPRC primary paired with the secondary being queried.

FORMAT(NO | FQMAP)
Specifies the format of the data returned in RETINFO.

NO Message format, delimited by commas. This is the default. For information on the data returned, see the description of messages ANTF0095I and ANTF0425I in *z/OS MVS System Messages, Vol 1 (ABA-AOM)*.

FQMAP
The data returned is in hexadecimal format, mapped by macro ANTFQMAP. It will need to be interpreted by the calling script or program and is not printable. If the RETINFO area is printed, (returned in variable RESULT) the printed data would appear as `'........'`.

The data returned is the EBCDIC hexadecimal interpretation of the data mapped by macro ANTFQMAP in variable RESULTX. This information will need to be interpreted by the calling script or program, but will contain printable characters. The sample ANTFREXX will display query information as in a dump formatter when QRYINFO returns hexadecimal data.

```
+00000000 C1D5E3C6 D8D4C1D7 01010000 00000000 000000BC 00000068 00000010 00000000 *ANTFQMAP....................
+00000020 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *............................
+00000040 0F450000 C8004000 F0F0F0F0 F0F0F0F0 F1F6F9F1 00000000 F2F1F0F7 A8000000 *....H. .000000001691....2107y
+00000060 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *............................
+00000080 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *............................
+000000A0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000002 *............................
+000000C0 2C000000 00000002 00000000 00402000 00000000 C800C800 00001DF3 00010009 *............ ......H.H....3.
+000000E0 02010009 00001DF3 4A09FEE1 4A09FEE1 00000000 00000000 0040A000 00000000 *.......3................ ...
+00000100 C800C800 00001DF3 00010009 02010009 00001DF3 4A09FEE1 4A09FEE1 00000000 *H.H....3...........3.........
+00000120 00000000 00000002 0000413B 00020200 00000000 *...................
```

WAITTIME(*waittime* | 0)
Specifies how long in seconds SDM will wait for a request to complete. A value of zero specifies that the request should not be timed.

If the time expires before the request is complete, the program returns a return code of 7039 (RQST_WAITTIME_EXPIRED). For more information about RQST_WAITTIME_EXPIRED, refer to the ANTRQSTL macro.

SUBCHSET(*subchset* | 0)
Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is supported by the host system's processor and configured for the device.

If you omit the parameter, the command is issued to the device that is logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

Return codes for QRYSIZE

If the length specified in QRYSIZE is not big enough to contain the complete result, the program returns a return code of 7622

(RQST_FCQUERY_QRYINFO_TOO_SMALL). The program places the total space required in the reason code. The truncated data is in the QRYINFO data area; however, RESULTX will be empty.

If the length specified in QRYSIZE is big enough to contain the complete result, the program returns a return code of 7623 (RQST_FCQUERY_QRYINFO_LARGE_ENOUGH). The program places the total amount of space that the result uses in the reason code part of RETINFO.

The return code is included in message ANTR8811E, which is returned in the RESULT variable.

Figure 72 shows how an exec might check the RESULT variable for the return code.

```
...

call anttrexx 'FCQUERY' , 'INOUT'

/* save the result variable */
presult = result

/* check for the error message ANTR8811E ERROR: request RETURN_CODE return_code  */
 parse var presult resulta resultb resultc resultd resulte resultrest
 if resulta = 'ANTR8811E' then
     do
          /* check for too-small return code */
          if resulte = 7622 then
             do
                 say 'QRYSIZE TOO SMALL, RESULTS TRUNCATED.'
             end

 ...
```

Figure 72. Example of Using the Return Code for QRYSIZE

FCWITHDRAW (FlashCopy withdraw)

Use request type FCWITHDRAW to end a FlashCopy relationship. This invokes ANTRQST ILK=ESSRVCS REQUEST=FCWITHDRAW.

SDEVN(*sdevn*)
Specifies the hexadecimal device number of a local CKD device that is the source of the relationship or relationships being withdrawn. SDEVN is mutually exclusive with REMOTE(YES) and OPENDVCS(YES).

TDEVN(*tdevn*)
Specifies hexadecimal device number of a local CKD device that is the target of the relationship or relationships being withdrawn. TDEVN is mutually exclusive with REMOTE(YES) and OPENDVCS(YES).

DEVN(*devn*)
Specifies the hexadecimal device number of a local CKD device that will be used as an access for an inband request (with REMOTE(YES) or fixed block request (with OPENDVCS(YES)).

If you use OPENDVCS(YES) without REMOTE(YES), DEVN must specify a CKD access volume located in the same subsystem cluster as the fixed block device identified by SOURCE in this command.

If you use OPENDVCS(YES) with REMOTE(YES), DEVN must specify a CKD access volume located in the same subsystem cluster as the PPRC primary device that is paired with the PPRC secondary specified as the FlashCopy source.

If you use REMOTE(YES) without OPENDVCS(YES), DEVN must specify the PPRC primary device that is paired with the PPRC secondary specified as the FlashCopy source.

SRCSERIAL(*srcserial*)
Specifies the 10-digit hexadecimal storage control serial number for the FlashCopy source. SRCSERIAL is required when DEVN is specified.

SRCLSS(*srclss*)
Specifies the 2-digit hexadecimal logical subsystem number for the FlashCopy source. SRCLSS is required when DEVN is specified.

SRCDVC(*srcdvc*)
Specifies the 2-digit hexadecimal value for the FlashCopy source device, either the connection address or the logical unit number. SRCDVC is required when DEVN is specified.

SRCSSID(*srcssid*)
When REMOTE(YES) is specified without OPENDVCS(YES), this specifies the 4-digit hexadecimal value for the subsystem where the FlashCopy is to be withdrawn. This must be the same value as that specified for SSID on the PPRC secondary volume on the PPRC establish pair command.

TGTSERIAL(*tgtserial*)
Specifies the 10-digit hexadecimal storage control serial number for the FlashCopy target. TGTSERIAL is required when either DEVN or TGTUCB(NO) is specified.

TGTLSS(*tgtlss*)
Specifies the 2-digit hexadecimal logical subsystem number for the FlashCopy target. TGTLSS is required when either DEVN or TGTUCB(NO) is specified.

TGTDVC(*tgtdvc*)
Specifies the 2-digit hexadecimal value for the FlashCopy target device, either the connection address or the logical unit number. TGTDVC is required when either DEVN or TGTUCB(NO) is specified.

SRCEXTNA()
Indicates that all tracks of the source device are to be withdrawn (this is a full volume request). Either SRCEXTNA or SRCEXTN is required with SDEVN. SRCEXTNA is mutually exclusive with SRCEXTN.

This corresponds to the ANTRQST keyword SRCEXTENTS with a value of ALL.

SRCEXTN(*srcextent*,*srcextent*,*srcextent*,*srcextent*)
Indicates track address ranges on the source CKD volume that are to be withdrawn. Each extent range consists of a starting track address and an ending track address. The tracks identified in the source extent fields must be valid for the source device.

Either SRCEXTNA or SRCEXTN is required with SDEVN. SRCEXTN is mutually exclusive with SRCEXTNA.

Supply the value for SRCEXTN in SUPERZAP format, with a comma between every 4 digits. For example, to copy source tracks 01110002 through 011100C, you would use SRCEXTN(0111,0002,0111,000C). To supply multiple source extent ranges, specify each range with a separate keyword. For instance, to copy source tracks 01110002 through 0111000C as well as tracks 02220003 through 0222000D, you would use this:

```
SRCEXTN(0111,0002,0111,000C)
SRCEXTN(0222,0003,0222,000D)
```

instead of this:

```
SRCEXTN(0111,0002,0111,000C,0222,0003,0222,000D)
```

You can supply up to 40 extent ranges on a single FCWITHDRAW request.

This corresponds to the ANTRQST keyword SRCEXTENTS.

TGTEXTNA()
Indicates that all tracks of the target device are to be withdrawn (this is a full volume request). Either TGTEXTNA or TGTEXTN is required with SDEVN. TGTEXTNA is mutually exclusive with TGTEXTN.

This corresponds to the ANTRQST keyword TGTEXTENTS with a value of ALL.

TGTEXTN(*nnnn,nnnn,nnnn,nnnn*)
Indicates track address ranges on the target CKD volume that is active in this FlashCopy relationship. Each extent range consists of a starting track address and an ending track address. The tracks identified in the target extent fields must be valid for the target device.

Either TGTEXTNA or TGTEXTN is required when TDEVN is specified. TGTEXTN is mutually exclusive with TGTEXTNA

Supply the value for TGTEXTN in SUPERZAP format, with a comma between every 4 digits. For example, to copy to target tracks 01110002 through 011100C, you would use TGTEXTN(0111,0002,0111,000C). To supply multiple source extent ranges, specify each range as a separate keyword. For instance, to copy to tracks 01110002 through 011100C as well as to tracks 02220003 through 0222000D, you would use this:

```
TGTEXTN(0111,0002,0111,000C)
SRCEXTN(0222,0003,0222,000D)
```

instead of this:

```
TGTEXTN(0111,0002,0111,000C,0222,0003,0222,000D)
```

You can supply up to 40 extent ranges on a single FCWITHDRAW request.

This corresponds to the ANTRQST keyword TGTEXTENTS with a value of ALL.

OPENDVCS(NO | YES)
Specifies whether Open System (fixed block) devices or CKD devices are addressed by this command.

NO The required parameters (SDEVN and TDEVN) identify CKD devices. This is the default.

YES
The required parameters (DEVN, SRCSERIAL, SRCLSS, SRCSSID, SRCDVC, TGTSERIAL, TGTLSS, TGTDVC) identify fixed block devices using a CKD access device.

REMOTE(NO | YES)
Specifies whether the request is for a subsystem that is not directly attached to the issuing processor.

NO The request is directed to a device on a subsystem locally attached to the issuing processor. This is the default.

YES
The request is directed to a remote subsystem using inband subsystem functions. When REMOTE(YES) is specified, the request must be for a full volume.

When specifying an inband request for a CKD secondary device, DEVN must identify a PPRC primary device in a subsystem accessible by the host processor issuing the request, and the FlashCopy source device must be the PPRC secondary of that DEVN.

When specifying an inband request for an Open System (fixed block) secondary device, DEVN must identify an online CKD device located in the same subsystem cluster as the fixed block device of the PPRC primary that is paired with the PPRC secondary that is designated as the FlashCopy source.

SPACEREL(NO | YES)
Specifies whether the physical space associated with a space efficient volume or a space efficient target is to be released.

NO The space is not to be released if the volume is space efficient or the target is space efficient. This is the default.

YES
The space associated with the target volume, if it is space efficient, is to be released when the withdraw is complete.

Note:

1. If YES is specified and the target is not a space efficient volume, the SPACEREL keyword is ignored. The results of the command causes the space efficient volume to become uninitialized and therefore unusable by the system until the volume is re-initialized.
2. If the request with SPACEREL(YES) is issued to a simplex volume that is a space efficient volume, any physical space for the volume is released.
3. The allocated space is released asynchronously so may not be reported immediately.

ACTION(COMMIT | REVERT | THAW)
Specifies an action to take. There is no default. If ACTION is specified, any extent specifications are ignored.

COMMIT
The last consistency group created by the Global Mirror session is committed to the target's current state, and reverting to the previous consistency group state is no longer possible.

REVERT
A rollback is performed to the state saved by a previous automatic FlashCopy establish command. The FlashCopy relationship is not removed (withdrawn); the FlashCopy target is rolled back to the previous consistency group created by the Global Mirror session.

THAW
A FlashCopy consistency group has been formed and application I/O can now take place on all volumes in the LSS where the previous FlashCopy Establish request was issued with the ACTION parameter specified as FREEZE. The DEVN parameter must be specified when ACTION(THAW) is specified. SDEVN and TDEVN cannot be used.

DDSW(NO | YES)
Specifies if the deleted dataspace withdraw function (DDSW) is requested. The DDSW function causes two actions. First, within the specified source extents, all FlashCopy target tracks are immediately withdrawn from their FlashCopy relationship. Second, all FlashCopy source tracks in a NOCOPY relationship are changed to a COPY (background copy) relationship. DDSW(YES) requires the source device, SDEVN, to be specified. If DDSW(YES) is specified, the target device parameters, TDEVN and target extent specifications, are ignored.

NO The DDSW function is not requested. This is the default.

YES
Background copy is started for all source tracks within the extents specified and any target relationships are immediately withdrawn.

TGTUCB(YES | NO)
Specifies whether an MVS device number is used to address the target.

YES
The SDEVN and TDEVN keywords are used to identify the source and target devices in the relationship. The MVS device numbers are used in the SDEVN and TDEVN keywords to address the source and target devices. This is the default.

NO The MVS device number is specified in the SDEVN keyword to identify the source. The serial number, LSS, and CCA are specified in the TGTSERIAL, TGTLSS, and TGTDVC keywords to address the target device.

TGTUCB(NO) is not applicable with OPENDVCS(YES) or REMOTE(YES).

WAITTIME(*waittime* | 0)
Specifies how long in seconds SDM will wait for a request to complete. A value of zero specifies that the request should not be timed.

If the time expires before the request is complete, the program returns a return code of 7039 (RQST_WAITTIME_EXPIRED). For more information about RQST_WAITTIME_EXPIRED, refer to the ANTRQSTL macro.

SUBCHSET(*subchset* | 0)
Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is supported by the host system's processor and configured for the device.

If you omit the parameter, the command is issued to the device that is logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

TSUBCHSET(*tsubchset* | 0)
Specifies which subchannel set is to be used to get information about the target device specified with the TDEVN parameter. This is the subchannel set for the device as defined in the Hardware Configuration Dialog (HCD). The values are determined by what is currently supported by the host system's processor and configured for the device.

If you omit the parameter, the device that is currently logically in subchannel set 0 will be used. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), via HCD.

Note: When a FlashCopy NOCOPY relationship is ended, the track data on the target device is unpredictable and should not be used. If updates occur to source device tracks in the FlashCopy NOCOPY relationship, a copy of the source tracks from the point-in-time of the FlashCopy establish may or may not be written to the target device.

Tracks may be copied from the source to the target volume even if the source track is not changed. This includes the track that contains the volume label. Therefore, to avoid duplicate volume serial problems when the target device is later varied online, IBM recommends that you label the target volume again after withdrawing a volume-level FlashCopy NOCOPY relationship.

LEVEL (query ANTRQST level)

Use request type LEVEL to query the level of ANTRQST that is installed on the system. This invokes ANTRQST ILK=*ilk* REQUEST=LEVEL.

ILK(*ilk***)**
: Specifies the ILK for the ANTRQST invocation. It is required. Valid values are:

  ESSRVCS
  : FlashCopy and Global Mirror

  PPRC
  : Peer-to-peer remote copy

  XRC
  : Extended remote copy

Refer to the prolog of ANTRQSTL for information about the level indicators.

QHA (Query where path groups are established)

Use request type QHA to query a device and determine where path groups are established. The device is considered online where path groups are established. This might be useful, for example, when a FlashCopy or Peer-to-Peer Remote Copy establish command fails due to an online target.

This invokes ANTRQST ILK=ESSRVCS REQUEST=QHA.

DEVN(*devn***)**
: Specifies the MVS device number to which the command is to be issued.

,CCA(*cca***)**
: Specifies the 2-digit hexadecimal value of the CCA for the device to be queried. It is required if you specify TGTUCB(NO) and ignored otherwise.

,LSS(*lss***)**
: Specifies the 2-digit hexadecimal value of the LSS that is to be queried. It is required if you specify TGTUCB(NO) and ignored otherwise.

QRYINFO()
: Creates the data area where the query information is to be returned. This area is the size specified with the QRYSIZE keyword. QRYINFO is required.

QRYSIZE(*qrysize***)**
: Specifies the decimal length of the QRYINFO area. It is required.

SUBCHSET(*subchset***)**
: Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in

the Hardware Configuration Dialog (HCD). The valid values are determined by what is supported by the host system's processor and configured for the device.

If you omit the parameter, the command is issued to the device that is logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

TGTUCB(YES | NO)
Specifies whether the query results are requested for a FlashCopy device with no target UCB. The values are:

YES
Indicates that the query results are requested for the device specified by DEVN. The LSS and CCA keywords are ignored. YES is the default.

NO Indicates that LSS and CCA keywords should be used to identify the device to be queried. You must also use the DEVN keyword to specify a device within the same SFI as the device to be queried.

FENCE (PPRC prevent unintended access through Soft Fence)

Use request type FENCE to prevent unintended access, through Soft Fence, of a device, a set of devices in an LSS or all devices in an LSS. This invokes ANTRQST ILK=PPRC REQUEST=FENCE.

DEVN(*dev*)
Specifies the device number to which the FENCE command is issued. This device number must identify a CKD device defined to the host system that is making the request. It is required.

ACTION(*action*)
Specifies an action to take.

| **FENCE**
| A Soft Fence operation is requested.

| **UNFENCE**
| A Reset Soft Fence operation is requested.

| **SPIDFENCE**
| A SPID Fence operation is requested.

| **SPIDUNFENCE**
| A Reset SPID Fence operation is requested.

DVC(*dvc*)
Indicates the binary logical unit number (LUN) for the device to be fenced or unfenced. Required if you specify OPENDVCS(YES) and SCOPE(DEV); ignored otherwise.

LSS(*lss*)
Specifies the binary logical subsystem number for the FB device to be fenced or unfenced. Required if you specify OPENDVCS(YES); ignored otherwise.

MASK(*mask*)
Specifies a 32-byte mask, each bit of which represents a device on which to perform a soft fence operation. Bit 0 of the first byte represents device 0. Bit 7 of the last byte represents device 255. MASK is required when you specify SCOPE(MASK). Otherwise, it is ignored.

OPENDVCS(NO | YES)
Indicates if the request is for Open System (fixed block) devices.

NO The device specified with DEVN is a CKD devices. This is the default.

YES
The request applies to an Open System LSS or device.

SCOPE(*scope*)
Specifies the scope of the fence operation. The values are:

DEV
The specified device only. This is the default.

LSS
All devices in the LSS of the device specified in the DEVN keyword or the LSS keyword if OPENDVCS(YES).

MASK
All devices specified by the MASK keyword. You must also specify MASK(*mask*) with a non-zero mask.

SUBCHSET(*subchset* | 0)
Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is supported by the host system's processor and configured for the device.

If you omit the parameter, the command is issued to the device that is logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

| **TOKEN(*token*)**
| Specifies an 8-byte field containing the host token. The token is a unique
| identifier used to identify the owner of the SPID Fence for the device. The
| TOKEN keyword is valid only for ACTION(SPIDFENCE) and
| ACTION(SPIDUNFENCE). The TOKEN is ignored for all other ACTION
| keyword values.)

PDELPAIR (PPRC delete volume pair)

Use request type PDELPAIR to delete a PPRC volume pair. This invokes ANTRQST ILK=PPRC REQUEST=PDELPAIR.

DEVN(*dev*)
Specifies the hexadecimal device number to be used for I/O. This device number must identify a CKD device defined to the host system that is making the request. It is required.

PCCA(*pcca*)
Specifies the hexadecimal channel connection address (CKD device CCA) or logical unit number (fixed block device LUN) of the primary device. It is required.

PLSS(*plss*)
Specifies the hexadecimal logical subsystem number (LSS) within the primary storage subsystem. PLSS is required for storage subsystems with logical subsystems and is invalid for 3990 and other 3990-compatible storage controls.

PSERIAL(*pserial*)
Specifies the EBCDIC serial number of the primary storage control. It is required.

PSSID(*pssid*)
Specifies the hexadecimal subsystem identifier (SSID) of the primary storage control. It is required.

SCCA(*scca*)
Specifies the hexadecimal channel connection address (CKD device CCA) or logical unit number (fixed block device LUN) of the secondary device. It is required.

SLSS(*slss*)
Specifies the hexadecimal logical subsystem number (LSS) within the secondary storage subsystem. SLSS is required for storage subsystems with logical subsystems and is invalid for 3990 and other 3990-compatible storage controls.

SSERIAL(*sserial*)
Specifies the EBCDIC serial number of the secondary storage control. It is required.

SSSID(*sssid*)
Specifies the hexadecimal subsystem identifier (SSID) of the secondary storage control. It is required.

OPENDVCS(NO | YES)
Indicates if the request is for Open System (fixed block) devices.

NO The device specified with DEVN, as well as the primary and secondary devices, are CKD devices. This is the default.

YES
The values supplied for the primary and secondary devices (SSID, Serial, LSS and LUN) apply to Open Systems devices and not the CKD DEVN device. The DEVN parameter identifies a CKD device used for receiving the request.

WAITTIME(*waittime* | 0)
Specifies the number of seconds that SDM must wait for a request to complete. A value of zero specifies that the request should not be timed. Valid values are 0 through 65535.

The default is 0.

SUBCHSET(*subchset* | 0)
Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is supported by the host system's processor and configured for the device.

If you omit the parameter, the command is issued to the device that is logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

PDELPATH (PPRC delete path)

Use request type PDELPATH to delete PPRC paths. This invokes ANTRQST ILK=PPRC REQUEST=PDELPATH.

DEVN(*devn***)**
Specifies the hexadecimal device number to be used for I/O. This device number must identify a CKD device defined to the host system that is making the request. It is required.

PLSS(*plss***)**
Specifies the hexadecimal logical subsystem number (LSS) within the primary storage subsystem. PLSS is required for storage subsystems with logical subsystems and is invalid for 3990 and other 3990-compatible storage controls.

PSERIAL(*pserial***)**
Specifies the EBCDIC serial number of the primary storage control. It is required. Specifying this parameter indicates that PPRC LINKs between the subsystems use ESCON communication protocols. PSERIAL requires the SSERIAL parameter. PSERIAL is mutually exclusive with PWWNN.

PWWNN(*pwwnn***)**
Specifies the hexadecimal world wide node name (WWNN) of the primary storage control. It is required. Specifying this parameter indicates that PPRC LINKs between the subsystems use Fibre Channel Protocol (FCP) communication protocols. PWWNN requires the SWWNN parameter. PWWNN is mutually exclusive with PSERIAL

PSSID(*pssid***)**
Specifies the hexadecimal subsystem identifier (SSID) of the primary storage control. It is required.

SLSS(*slss***)**
Specifies the hexadecimal logical subsystem number (LSS) within the secondary storage subsystem. SLSS is required for storage subsystems with logical subsystems and is invalid for 3990 and other 3990-compatible storage controls.

SSERIAL(*sserial***)**
Specifies the EBCDIC serial number of the secondary storage control. It is required. Specifying this parameter indicates that PPRC LINKs between the subsystems use ESCON communication protocols. SSERIAL requires the PSERIAL parameter. SSERIAL is mutually exclusive with. SWWNN.

SWWNN(*swwnn***)**
Specifies the hexadecimal world wide node name (WWNN) of the secondary storage control. It is required. Specifying this parameter indicates PPRC LINKs between the subsystems use Fibre Channel Protocol (FCP) communication protocols. SWWNN requires the PWWNN parameter. SWWNN is mutually exclusive with SSERIAL.

SSSID(*sssid***)**
Specifies the hexadecimal subsystem identifier (SSID) of the secondary storage control. It is required.

OPENDVCS(NO | YES)
Indicates if the request is for Open System (fixed block) devices.

NO The device specified with DEVN, as well as the primary and secondary devices, are CKD devices. This is the default.

YES
The values supplied for the primary and secondary devices (SSID, Serial, LSS and LUN) apply to Open Systems devices and not the CKD DEVN device. The DEVN parameter identifies a CKD device used for receiving the request.

WAITTIME(*waittime* | 0)
Specifies the number of seconds that SDM must wait for a request to complete. A value of zero specifies that the request should not be timed. Valid values are 0 through 65535.

The default is 0.

SUBCHSET(*subchset* | 0)
Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is supported by the host system's processor and configured for the device.

If you omit the parameter, the command is issued to the device that is logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

PESTPAIR (PPRC establish volume pair)

Use request type PESTPAIR to establish a PPRC volume pair. This invokes ANTRQST ILK=PPRC REQUEST=PESTPAIR.

DEVN(*devn*)
Specifies the hexadecimal device number to be used for I/O. This device number must identify a CKD device defined to the host system making the request. It is required.

PCCA(*pcca*)
Specifies the hexadecimal channel connection address (CKD device CCA) or logical unit number (fixed block device LUN) of the primary device. It is required.

PLSS(*plss*)
Specifies the hexadecimal logical subsystem number (LSS) within the primary storage subsystem. PLSS is required for storage subsystems with logical subsystems and is invalid for 3990 and other 3990-compatible storage controls.

PSERIAL(*pserial*)
Specifies the EBCDIC serial number of the primary storage control. It is required.

PSSID(*pssid*)
Specifies the hexadecimal subsystem identifier (SSID) of the primary storage control. It is required.

SCCA(*scca*)
Specifies the hexadecimal channel connection address (CKD device CCA) or logical unit number (fixed block device LUN) of the secondary device. It is required.

SLSS(*slss*)
Specifies the hexadecimal logical subsystem number (LSS) within the secondary storage subsystem. SLSS is required for storage subsystems with logical subsystems and is invalid for 3990 and other 3990-compatible storage controls.

SSERIAL(*sserial*)
Specifies the EBCDIC serial number of the secondary storage control. It is required.

SSSID(*sssid*)
Specifies the hexadecimal subsystem identifier (SSID) of the secondary storage control. It is required.

ACTION(FAILOVER | FAILBACK| MTFAILOVER)
Specifies that an action is to be taken on an already existing volume pair. This is mutually exclusive with MODE(INCRES) and INCRESYNC(START | STOP).

FAILOVER
The direction of the PPRC pair is to be reserved and change recording is to begin. Upon completion of the command, the specified primary device will be a suspended primary of a PPRC pair. Be sure to reverse devices when specifying primary and secondary values.

FAILBACK
The failure condition at the primary site has been resolved and that the changes that have taken place at the recovery site since the FAILOVER command was issued should now be copied to the original primary site.

MTFAILOVER
Specifies that a multi-target configuration should be created from a cascaded configuration. For example, in a cascaded configuration, H1->H2->H3, where H1 is the primary, issuing a CESTPAIR command with ACTION(MTFAILOVER) to the middle device, H2, causes a multi-target configuration (H2->H1 and H2-H3) in place of the previous cascaded failover state.

For more information, see “Failover/Failback in a PPRC environment” on page 361.

AUTORESYNC(YES | NO)
Specifies whether the auto resynchronization capability is to be enabled for a global copy pair.

NO Auto resynchronization is disabled for this volume pair

YES
Auto resynchronization is enabled for this volume pair. This is the default.

CASCADE(NO | YES)
Indicates this request should identify the devices as part of a cascading PPRC pair.

NO Primary and secondary devices are marked as not cascading capable. This is the default.

YES
Primary and secondary devices are marked as cascading capable

CRIT(NO | YES)
Indicates whether this is a critical volume.

NO If an I/O error occurs, subsequent write requests to the PPRC primary device are allowed. This is the default.

YES
If an I/O error occurs, writes to the primary device are either allowed or not allowed, based on the 3990 maintenance panel VPD bit setting or the CGROUP selection on the PPRC establish pair command

CRIT is mutually exclusive with OPTION(XD).

FORCE(NO | YES)
Specifies whether validation of the volumes involved in a MODE(INCRES) should occur or be bypassed.

NO Fail processing if the secondary device is online to a host system. This is the default.

YES
Continue establish pair processing when the secondary device is online to a host system

This parameter is valid only when MODE(INCRES) is specified.

INCRESYNC(START | STOP | NOCOPY)
Specifies whether the incremental resync change recording mechanism for Metro/Global Mirror should be enabled.

START
Start the change recording mechanism

STOP
Stop the change recording mechanism

NOCOPY
Used in recovery scenarios, when the incremental resync change recording mechanism needs to be started. The assumption is that the local and remote volumes involved in the cascading relationship are identical; therefore, the bitmaps used are not initialized with a full copy.

INCRESYNC(START) and INCRESYNC(STOP) are mutually exclusive with ACTION(FAILOVER), ACTION(FAILBACK) and MODE(INCRES).

MODE(COPY | NOCOPY | RESYNC | INCRES)
Specifies how a volume pair is to be set up.

COPY
Copy all tracks on the primary volume to the secondary volume. This is the default.

NOCOPY
Copy to the secondary volume only those tracks on the primary volume that are updated after this request is processed

RESYNC
Reestablish a suspended copy and copy to the secondary volume any tracks updated since the suspension

INCRES
Requests an incremental resynchronization using the change recording bitmaps. MODE(INCRES) is mutually exclusive with ACTION(FAILOVER | FAILBACK) and INCRESYNC(START | STOP).

MSGREQ(NO | YES)
Indicates whether this request should wait for the initial copy of a volume to complete.

NO Processing should continue without waiting for the initial volume copy to complete. This is the default.

YES
Processing should wait for the initial volume copy to complete.

ONLINSEC(NO | YES)
Specifies whether establish pair processing should continue if the secondary device is online to a host system.

NO Fail processing if the secondary device is online to a host system. This is the default.

YES
Continue establish pair processing when the secondary device is online to a host system

OPENDVCS(NO | YES)
Indicates if the request is for Open System (fixed block) devices.

NO The device specified with DEVN, as well as the primary and secondary devices, are CKD devices. This is the default.

YES
The values supplied for the primary and secondary devices (SSID, Serial, LSS and LUN) apply to Open Systems devices and not the CKD DEVN device. The DEVN parameter identifies a CKD device used for receiving the request.

OPTION(SYNC | XD)
Indicates whether PPRC should copy tracks to the secondary volume prior to updating the primary volume.

SYNC
Changed tracks should be copied to the secondary volume before the updates are written to the primary volume. This is the default.

XD PPRC should record track changes to the primary volume in a bitmap. At intervals determined by the subsystem, PPRC will copy all changed primary volume tracks to the secondary volume. The tracks are copied in bitmap order, not necessarily in the order they were updated on the primary volume.

PACE(*pace*)
Specifies the minimum number of tracks to be copied at one time before allowing another host to interrupt. This parameter is used only for 3990 storage controls.

RESETRSV(NO | YES)
Specifies whether an existing reserve on the specified fixed block secondary device should be reset (removed) by the control unit during the establish pair processing.

NO Any reserve that exists on the specified secondary device should not be reset. This is the default.

YES
If there is a reserve on the specified secondary device, that reserve should be reset.

SETGTOK(NO | YES)
Specifies whether a secondary can be a space efficient volume for this request.

NO A space efficient volume will not be used as a PPRC secondary device. This is the default.

YES
A space efficient volume may be used as a PPRC secondary device on this request

WAITTIME(*waittime* | 0)
Specifies the number of seconds that SDM must wait for a request to complete. A value of zero specifies that the request should not be timed. Valid values are 0 through 65535. The default is 0.

SUBCHSET(*subchset* | 0)
Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is supported by the host system's processor and configured for the device.

If you omit the parameter, the command is issued to the device that is logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

PESTPATH (PPRC establish path)

Use request type PESTPATH to establish PPRC paths. This invokes ANTRQST ILK=PPRC REQUEST=PESTPATH.

DEVN(*devn*)
Specifies the hexadecimal device number to be used for I/O. This device number must identify a CKD device defined to the host system making the request. It is required.

PLSS(*plss*)
Specifies the hexadecimal logical subsystem number (LSS) within the primary storage subsystem. PLSS is required for storage subsystems with logical subsystems and is invalid for 3990 and other 3990-compatible storage controls.

PSERIAL(*pserial*)
Specifies the EBCDIC serial number of the primary storage control. It is required. Specifying this parameter indicates that PPRC LINKs between the subsystems use ESCON communication protocols. PSERIAL requires SSERIAL and is mutually exclusive with PWWNN.

PWWNN(*pwwnn*)
Specifies the hexadecimal world wide node name (WWNN) of the primary storage control. It is required. Specifying this parameter indicates that PPRC LINKs between the subsystems use Fibre Channel Protocol (FCP) communication protocols. PWWNN requires the SWWNN parameter and is mutually exclusive with PSERIAL

PSSID(*pssid*)
Specifies the hexadecimal subsystem identifier (SSID) of the primary storage control. It is required.

SLSS(*slss*)
Specifies the hexadecimal logical subsystem number (LSS) within the secondary storage subsystem. SLSS is required for storage subsystems with logical subsystems and is invalid for 3990 and other 3990-compatible storage controls.

SSERIAL(*sserial*)
Specifies the EBCDIC serial number of the secondary storage control. It is required. Specifying this parameter indicates that PPRC LINKs between the subsystems use ESCON communication protocols. SSERIAL requires the PSERIAL parameter and is mutually exclusive with SWWNN

SWWNN(*swwnn*)
Specifies the hexadecimal world wide node name (WWNN) of the secondary storage control. It is required. Specifying this parameter indicates that PPRC LINKs between the subsystems use Fibre Channel Protocol (FCP) communication protocols. SWWNN requires the PWWNN parameter and is mutually exclusive with SSERIAL.

SSSID(*sssid*)
Specifies the hexadecimal subsystem identifier (SSID) of the secondary storage control. It is required.

LINKS(*link1*,*link2*,*link3...*)
Specifies the hexadecimal path address from the application site storage control to the recovery site storage control. You can specify up to 4 hexadecimal path addresses, separated by commas, with each LINKS keyword, and you can specify LINKS multiple times. For example, you could specify the following for up to 8 path addresses per PESTPATH command:

```
LINKS(11111111,22222222,33333333)
LINKS(44444444,55555555,66666666)
```

LINKS is mutually exclusive with LINK1. LINKS or LINK1 are required.

LINK1(*link1*)
Specifies the hexadecimal path address from the application site storage control to the recovery site storage control. LINK1 is mutually exclusive with LINKS. You must specify either LINKS or LINK1.

LINK2(*link2*)
Specifies the hexadecimal path address from the application site storage control to the recovery site storage control. LINK2 requires the LINK1 parameter.

LINK3(*link3*)
Specifies the hexadecimal path address from the application site storage control to the recovery site storage control. LINK3 requires the LINK2 parameter.

LINK4(*link4*)
Specifies the hexadecimal path address from the application site storage control to the recovery site storage control. LINK4 requires the LINK3 parameter.

CGROUP(NO | YES)
Indicates a logical control unit subsystem should emulate 3990 Critical Volume mode for CGROUP automation.

NO 3990 Critical Volume mode should not be emulated. This is the default.

YES
3990 Critical Volume mode should be emulated, permitting CGROUP automation processing of PPRC pair errors

RESETHP(NO | YES)
Specifies whether to reset host path connections to the primary subsystem before attempting to establish paths.

NO Host path connections to the primary subsystem will not be reset. The request will return an error indicating some or all paths failed. This is the default

YES
Host paths to the primary subsystem on the specified link addresses should be reset to permit the request to use the link addresses for connections to the secondary subsystem.

OPENDVCS(NO | YES)
Indicates if the request is for Open System (fixed block) devices.

NO The device specified with DEVN, as well as the primary and secondary devices, are CKD devices. This is the default.

YES
The values supplied for the primary and secondary devices (SSID, Serial, LSS and LUN) apply to Open Systems devices and not to the CKD DEVN device. The DEVN parameter identifies a CKD device used for receiving the request.

WAITTIME(*waittime* | 0)
Specifies the number of seconds that SDM must wait for a request to complete. A value of zero specifies that the request should not be timed. Valid values are 0 through 65535.

The default is 0.

SUBCHSET(*subchset* | 0)
Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is supported by the host system's processor and configured for the device.

If you omit the parameter, the command is issued to the device that is logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

PFREEZE (PPRC control volume groups)

Use request type PFREEZE to perform a PPRC freeze request (PPRC CGROUP). This invokes ANTRQST ILK=PPRC REQUEST=PFREEZE. For more information, see "CGROUP – controlling volume groups" on page 308.

DEVN(*devn*)
Specifies the hexadecimal device number to be used for I/O. It is required. The device number must identify a CKD device that is defined to the host system making the request.

PLSS(*plss*)
Specifies the hexadecimal logical subsystem number (LSS) within the primary storage subsystem. PLSS is required for storage subsystems with logical subsystems and is invalid for 3990 and other 3990-compatible storage controls.

PSERIAL(*pserial*)
Specifies the EBCDIC serial number of the primary storage control. It is required.

PSSID(*pssid*)
Specifies the hexadecimal subsystem identifier (SSID) of the primary storage control. It is required.

SLSS(*slss*)
Specifies the hexadecimal logical subsystem number (LSS) within the secondary storage subsystem. SLSS is required for storage subsystems with logical subsystems and is invalid for 3990 and other 3990-compatible storage controls.

SSERIAL(*sserial*)
Specifies the EBCDIC serial number of the primary storage control. It is required.

SSSID(*sssid*)
Specifies the hexadecimal subsystem identifier (SSID) of the secondary storage control. It is required.

OPENDVCS(NO | YES)
Indicates if the request is for Open System (fixed block) devices.

NO The device specified with DEVN, as well as the primary and secondary devices, are CKD devices. This is the default.

YES
The values supplied for the primary and secondary devices (SSID, Serial, LSS and LUN) apply to Open Systems devices and not to the CKD DEVN device. The DEVN parameter identifies a CKD device used for receiving the request.

WAITTIME(*waittime* | 0)
Specifies the number of seconds that SDM must wait for a request to complete. A value of zero specifies that the request should not be timed. Valid values are 0 through 65535. The default is 0.

SUBCHSET(*subchset* | 0)
Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is supported by the host system's processor and configured for the device.

If you omit the parameter, the command is issued to the device that is logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

PQSCSTAT (Query storage controller)

Use request type PQSCSTAT to query storage controller status. This invokes ANTRQST ILK=PPRC REQUEST=PQSCSTAT.

DEVN(*devn*)
Specifies the hexadecimal device number to be used for I/O. It is required. The device must be located in the same subsystem cluster as the device specified in the QRYLSS and QRYDVC parameters.

QRYLSS(*qrylss*)
Specifies the hexadecimal logical subsystem number (LSS). QRYLSS is required.

QRYSIZE(*qrysize*)
Specifies the decimal length of the QRYINFO area. It is required.

QRYINFO(*qryinfo*)
Creates the data area where the query information is to be returned. The returned data is mapped by mapping macro (DSECT) ANTPQSCS.

OPENDVCS(NO | YES)
Indicates if the request is for Open System (Fixed Block) devices. It is optional.

NO The query applies to a CKD LSS or device. This is the default.

YES
The query applies to an Open System LSS or device.

QRYDVC(*qrydvc*)
Specifies the hexadecimal channel connection address (CCA) for the device that is to be queried. This parameter is required with SCOPE(RANK); otherwise, it is ignored.

SCOPE(LSS | RANK)
Specifies the content of the PQSCSTAT output. It is optional. The default is LSS.

LSS
Device and LSS summary information only is to be returned.

RANK
Device-specific rank information as well as the LSS and device summary information are to be returned.

WAITTIME(*waittime* | 0)
Specifies the number of seconds that SDM must wait for a request to complete. A value of zero specifies that the request should not be timed. Valid values are 0 through 65535. The default is 0.

SUBCHSET(*subchset* | 0)
Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is supported by the host system's processor and configured for the device.

If you omit the parameter, the command is issued to the device that is logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

PQUERY (PPRC query)

Use request type PQUERY to query PPRC status. This invokes ANTRQST ILK=PPRC REQUEST=PQUERY.

DEVN(*devn*)
Specifies the hexadecimal device number to be used for I/O. It is required. This device number must identify a CKD device that is defined to the host system making the request.

QRYSIZE(*qrysize*)
Specifies the decimal length of the QRYINFO area. It is required.

QRYINFO(*qryinfo*)
Creates the data area where the query information is to be returned.

QRYSERIAL(*qryserial*)
Specifies the EBCDIC serial number of the storage control being queried. QRYSERIAL is required when OPENDVCS(YES) is specified.

QRYLSS(*qrylss*)
Specifies the hexadecimal logical subsystem number (LSS) that contains the Open System device being queried. QRYLSS is required when OPENDVCS(YES) is specified.

QRYDVC(*qrydvc*)
Specifies the hexadecimal logical unit number (LUN) of the Open System device being queried. QRYDVC is required when OPENDVCS(YES) is specified.

OPENDVCS(NO | YES)
Indicates if the request is for Open System (fixed block) devices.

NO The device specified with DEVN, as well as the primary and secondary devices, are CKD devices. This is the default.

YES
The values supplied for the primary and secondary devices (SSID, Serial, LSS and LUN) apply to Open Systems devices and not to the CKD DEVN device. The DEVN parameter identifies a CKD device used for receiving the request.

FORMAT(LONG | PQMAP)
Specifies the content of the PQUERY output.

LONG
All fields in the PQUERY output will be filled with relevant information as available. This is the default.

PQMAP
The data is not to be formatted. The sample ANTPREXX will display query information as in a dump formatter when QRYINFO returns hexadecimal data.

For more information, see FORMAT.

BITMAP(YES | NO)
Specifies whether the bitmap for the primary device should be accessed when determining the percentage of synchronization for a pending or suspended pair.

YES
PPRC will read the bitmap for the primary volume that is specified by DEVN, or by QRYDVC when OPENDVCS(YES) is also specified. This is the default.

NO PPRC will not read the bitmap

PATHS(NO | YES | LNK)
Specifies whether the query should display the pathing information for the pair that is queried.

NO PPRC will display the status of the volume specified by DEVN, or by QRYDVC when OPENDVCS(YES) is also specified. This is the default.

YES
PPRC will display all of the paths associated with the application site storage control, and the status of each path

LNK
PPRC displays all of the potential connectivity of the Fibre Channel ports in the storage control specified by the DEVN parameter to each system adapter port in the storage control specified by the SWWNN or SDEVNUM parameter. If a device specified by SDEVNUM is the secondary of an established PPRC pair, the query will fail. The exception to this rule is if the secondary device is also the primary device of another PPRC pair (cascading). Either SWWNN or SDEVNUM is required.

SWWNN(*swwnn***)**
Specifies the hexadecimal world wide node name (WWNN) of the secondary storage control. SWWNN is mutually exclusive with SDEVNUM.

SDEVNUM(*sdevnum***)**
Specifies a hexadecimal device number in the secondary storage control. SDEVNUM is mutually exclusive with SWWNN.

WAITTIME(*waittime* **| 0)**
Specifies the number of seconds that SDM must wait for a request to complete. A value of zero specifies that the request should not be timed. Valid values are 0 through 65535. The default is 0.

SUBCHSET(*subchset* **| 0)**
Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is supported by the host system's processor and configured for the device.

If you omit the parameter, the command is issued to the device that is logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

SSUBCHSET (*ssubchset* **| 0)**
Specifies the subchannel set for the secondary device specified with the SDEVN parameter. The subchannel set is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The values are determined by what is supported by the host system's processor and configured for the device.

If you omit the parameter, the command is issued to the device that is logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.)

| **QUIESCEDIO(YES | NO)**
| Specifies whether the request will be re-queued in the case of a
| device-not-ready condition caused by the quiesce processing.

| **YES**
| Requeues the request. This is the default.

| **NO** Immediately posts and fails the request.

PRECOVER (PPRC recover data on the recovery system)

Use request type PRECOVER to recover data on the recovery system. This invokes ANTRQST ILK=PPRC REQUEST=PRECOVER.

DEVN(*devn***)**
Specifies the hexadecimal device number to be used for I/O. It is required. This device number must identify a CKD device defined to the host system that is making the request.

PCCA(*pcca***)**
Specifies the hexadecimal channel connection address (CKD device CCA) or logical unit number (fixed block device LUN) of the primary device. It is required.

PLSS(*plss***)**
Specifies the hexadecimal logical subsystem number (LSS) within the primary

storage subsystem. PLSS is required for storage subsystems with logical subsystems and is invalid for 3990 and other 3990-compatible storage controls.

PSERIAL(*pserial***)**
Specifies the EBCDIC serial number of the primary storage control. It is required.

PSSID(*pssid* **)**
Specifies the hexadecimal subsystem identifier (SSID) of the primary storage control. It is required.

SCCA(*scca***)**
Specifies the hexadecimal channel connection address (CKD device CCA) or logical unit number (fixed block device LUN) of the secondary device. It is required.

SLSS(*slss***)**
Specifies the hexadecimal logical subsystem number (LSS) within the secondary storage subsystem. SLSS is required for storage subsystems with logical subsystems and is invalid for 3990 and other 3990-compatible storage controls.

SSERIAL(*sserial***)**
Specifies the EBCDIC serial number of the secondary storage control. It is required.

SSSID(*sssid***)**
Specifies the hexadecimal subsystem identifier (SSID) of the secondary storage control. It is required.

OPENDVCS(NO | YES)
Indicates if the request is for Open System (fixed block) devices.

NO The device specified with DEVN, as well as the primary and secondary devices, are CKD devices. This is the default.

YES
The values supplied for the primary and secondary devices (SSID, Serial, LSS and LUN) apply to Open Systems devices and not to the CKD DEVN device. The DEVN parameter identifies a CKD device used for receiving the request.

OLDVOLSER(*oldvolser***)**
Specifies the old volume serial number. PPRC verifies the OLDVOLSER contains characters and compares the device volume serial number to the OLDVOLSER value. If the volume serial numbers are equal, PPRC resets the volume to simplex state. OLDVOLSER is required if you specify NEWVOLSER.

NEWVOLSER(*newvolser***)**
Specifies the new volume serial number. If you specify NEWVOLSER, you must specify OLDVOLSER. PPRC verifies the OLDVOLSER contains characters and compares the device volume serial number to the OLDVOLSER value. If the volume serial numbers are equal, PPRC resets the volume to simplex state and replaces the old volume serial number with the new volume serial number.

WAITTIME(*waittime* **| 0)**
Specifies the number of seconds that SDM must wait for a request to complete. A value of zero specifies that the request should not be timed. Valid values are 0 through 65535. The default is 0.

SUBCHSET(*subchset* **|** **0)**
Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is supported by the host system's processor and configured for the device.

If you omit the parameter, the command is issued to the device that is logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

PRUN (PPRC control volume group)

Use request type PRUN to perform a PPRC run. This invokes ANTRQST ILK=PPRC REQUEST=PRUN. For more information, see RUN parameter at "CGROUP – controlling volume groups" on page 308.

DEVN(*devn***)**
Specifies the hexadecimal device number to be used for I/O. It is required. This device number must identify a CKD device defined to the host system making the request.

PLSS(*plss***)**
Specifies the hexadecimal logical subsystem number (LSS) within the primary storage subsystem. PLSS is required for storage subsystems with logical subsystems and is invalid for 3990 and other 3990-compatible storage controls.

PSERIAL(*pserial***)**
Specifies the EBCDIC serial number of the primary storage control. It is required.

PSSID(*pssid***)**
Specifies the hexadecimal subsystem identifier (SSID) of the primary storage control. It is required.

SLSS(*slss***)**
Specifies the hexadecimal logical subsystem number (LSS) within the secondary storage subsystem. SLSS is required for storage subsystems with logical subsystems and is invalid for 3990 and other 3990-compatible storage controls.

SSERIAL(*sserial***)**
Specifies the EBCDIC serial number of the primary storage control. It is required.

SSSID(*sssid***)**
Specifies the hexadecimal subsystem identifier (SSID) of the secondary storage control. It is required.

OPENDVCS(NO | YES)
Indicates if the request is for Open System (fixed block) devices.

NO The device specified with DEVN, as well as the primary and secondary devices, are CKD devices. This is the default.

YES
The values supplied for the primary and secondary devices (SSID, Serial, LSS and LUN) apply to Open Systems devices and not the CKD DEVN device. The DEVN parameter identifies a CKD device used for receiving the request.

WAITTIME(*waittime* **| 0)**
Specifies the number of seconds that SDM must wait for a request to complete. A value of zero specifies that the request should not be timed. Valid values are 0 through 65535. The default is 0.

SUBCHSET(*subchset* **| 0)**
Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is supported by the host system's processor and configured for the device.

If you omit the parameter, the command is issued to the device that is logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

PSETCHAR (PPRC set characteristics of volume pair)

Use request type PSETCHAR to set the characteristics of a PPRC volume pair, including whether the pair is to be used for a Preserve Mirror function in a Multi-Target Mirror configuration. This invokes ANTRQST ILK=PPRC REQUEST=PSETCHAR.

DEVN(*devn***)**
Specifies the binary device number to be used for I/O. It is required. If OPENDVCS=YES is specified, then *devn* identifies a System z device to act as a CKD access volume. This device must be located in the same subsystem cluster as the primary device..

PCCA(*pcca***)**
Slepcifies the binary channel connection address of the primary device. It is required. For open (fixed block) devices, this specifies the logical unit number (LUN).

PSERIAL(*pserial***)**
Specifies the EBCDIC serial number of the primary storage control. It is required.

PLSS(*plss***)**
Specifies the binary logical subsystem number (LSS) within the primary storage subsystem. It is required.

PSSID(*pssid***)**
Specifies the hexadecimal subsystem identifier (SSID) of the primary storage control. It is required.

SCCA(*scca***)**
Specifies the binary channel connection address of the secondary device. It is requried. For open (fixed block) devices, this specifies the logical unit number (LUN).

SSERIAL(*sserial***)**
Specifies the EBCDIC serial number of the secondary storage control. It is required.

SLSS(*slss***)**
Specifies the binary logical subsystem number (LSS) within the secondary storage subsystem. SLSS is required.

OPENDVCS(NO | YES)
Indicates if the request is for Open System (fixed block) devices.

NO Indicates the information for the primary and secondary devices (SSID, Serial, LSS, and CCA) applies to Open System devices, not the CKD DEVN device. The CCA value is treated as a Fixed Block Logical Unit Number (LUN). The DEVN parameter identifies a CKD device for receiving the request, but is not affected by this PPRC request.

YES
Is the same as omitting the OPENDVCS keyword. This value indicates the DEVN device and primary and secondary devices are CKD devices. This is the default.

USEFORPM(NO | YES)
Specifies whether the indicated pair is to be used for a Preserve Mirror function in a Multi-Target Mirror configuration. One and only one pair from a PPRC primary can have this enabled or Preserve Mirror Required (Remote Pair FlashCopy) functions will fail due to ambiguous settings.

NO This pair is not to be used for a Preserve Mirror function in a Multi-Target Mirror configuration. This is the default.

YES
This pair is to be used for a Preserve Mirror function in a Multi-Target Mirror configuration.

WAITTIME(*waittime* | 0)
Specifies how long, in seconds, SDM waits for a request to complete. A value of zero indicates that the system does not time the request. If the time expires before the request is complete a value of 7039 (RQST_WAITTIME_EXPIRED, refer to ANTRQSTL) or 7200 (RQST_PESTPAIR_ERROR, with a message ANTP0217E indicating I/O timeout) is returned. If the request is a synchronous request, the value is placed in the return code part of RETINFO.

SUBCHSET(*subchset*)
subchset is a 1-character field that specifies the subchannel set for the device where the command is to be issued. This is the subchannel set as defined in the Hardware Configuration Dialog (HCD). Valid values are:

0 Subchannel set 0. This is the default.

1 Subchannel set 1

2 Subchannel set 2

3 Subchannel set 3.

If you omit this keyword, the command is issued to the default device that is logically in subchannel set 0. If a swap has occurred, this could be the device that was physically defined in an alternate subchannel set. This is the default.

PSUSPEND (PPRC suspend volume pair)

Use request type PSUSPEND to suspend PPRC volume pairs. This invokes ANTRQST ILK=PPRC REQUEST=PSUSPEND.

DEVN(*devn*)
Specifies the hexadecimal device number to be used for I/O. It is required. This device number must identify a CKD device defined to the host system making the request.

PLSS(*plss*)
Specifies the hexadecimal logical subsystem number (LSS) within the primary

storage subsystem. PLSS is required for storage subsystems with logical subsystems and is invalid for 3990 and other 3990-compatible storage controls.

PSERIAL(*pserial***)**
Specifies the EBCDIC serial number of the primary storage control. It is required.

PSSID(*pssid***)**
Specifies the hexadecimal subsystem identifier (SSID) of the primary storage control. It is required.

SLSS(*slss***)**
Specifies the hexadecimal logical subsystem number (LSS) within the secondary storage subsystem. SLSS is required for storage subsystems with logical subsystems and is invalid for 3990 and other 3990-compatible storage controls.

SSERIAL(*sserial***)**
Specifies the EBCDIC serial number of the primary storage control. It is required.

SSSID(*sssid***)**
Specifies the hexadecimal subsystem identifier (SSID) of the secondary storage control. It is required.

OPENDVCS(NO | YES)
Indicates if the request is for Open System (fixed block) devices.

NO The device specified with DEVN, as well as the primary and secondary devices, are CKD devices. This is the default.

YES
The values supplied for the primary and secondary devices (SSID, Serial, LSS and LUN) apply to Open Systems devices and not to the CKD DEVN device. The DEVN parameter identifies a CKD device used for receiving the request.

WAITTIME(*waittime* **| 0)**
Specifies the number of seconds that SDM must wait for a request to complete. A value of zero specifies that the request should not be timed. Valid values are 0 through 65535. The default is 0.

SUBCHSET(*subchset* **| 0)**
Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is supported by the host system's processor and configured for the device.

If you omit the parameter, the command is issued to the device that is logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

RQUERY (Global Mirror query)

Use request type RQUERY to query a Global Mirror session. This invokes ANTRQST ILK=ESSRVCS REQUEST=RQUERY.

VOLSER(*volser***)**
Specifies the volume serial number of the device to be used for I/O. VOLSER is mutually exclusive with DEVN. Either VOLSER or DEVN is required.

DEVN(*devn*)
Specifies the hexadecimal device number to be used for I/O. DEVN is mutually exclusive with VOLSER. Either VOLSER or DEVN is required.

ACTION(*type*)
Specifies the query type desired for the request. It is required. The types are:

STAT4ALSS
Provides performance counters, consistency group information, FlashCopy information and out-of-sync tracks for the specified session within the specified LSS. The data is not to be formatted. The sample ANTRREXX will display query information as in a dump formatter when QRYINFO returns hexadecimal data.

STAT4AESS
Provides performance counters, consistency group information, FlashCopy information and out-of-sync tracks for the specified session within the ESS containing the specified LSS. The data is not to be formatted. The sample ANTRREXX will display query information as in a dump formatter when QRYINFO returns hexadecimal data.

STAT4ACGRP
Provides performance counters, consistency group information, FlashCopy information and out-of-sync tracks for the specified session within the ESS containing the specified LSS. The data is not to be formatted. The sample ANTRREXX will display query information as in a dump formatter when QRYINFO returns hexadecimal data.

STAT4BLSS
Provides out-of-sync information for the specified session within the specified LSS. The data is not to be formatted. The sample ANTRREXX will display query information as in a dump formatter when QRYINFO returns hexadecimal data.

STAT4BESS
Provides out-of-sync information for the specified session within the ESS containing the specified LSS. The data is not to be formatted. The sample ANTRREXX will display query information as in a dump formatter when QRYINFO returns hexadecimal data.

STAT4C
Provides summary information on all sessions within an ESS. The data is not to be formatted. The sample ANTRREXX will display query information as in a dump formatter when QRYINFO returns hexadecimal data.

STAT51
Provides detailed performance on all sessions within an ESS. The data is not to be formatted. The sample ANTRREXX will display query information as in a dump formatter when QRYINFO returns hexadecimal data.

DVCSTAT
Provides detailed status information on devices in a formatted output

GMPSTAT
Provides detailed status information on Global Mirror physical sessions in a formatted output

GMLSTAT
Provides detailed status information on Global Mirror logical sessions in a formatted output

QRYSIZE(*qrysize***)**
Specifies the decimal length of the QRYINFO area. It is required.

QRYINFO(*qryinfo***)**
Creates the data area where the query information is to be returned. This area is the size specified with the QRYSIZE parameter. QRYINFO is mutually exclusive with DSNAME. Either QRYINFO or DSNAME is required.

DSNAME(*dataset-name***)**
Specifies that XRC direct the RQUERY output to a sequential data set that is identified by *dataset-name.* The data set must be large enough to accommodate the requested RQUERY reports. DSNAME is mutually exclusive with QRYINFO. Either DSNAME or QRYINFO is required.

DSDISP(OLD | MOD | SHR)
Specifies how XRC writes the RQUERY output to the data set:

OLD
Clear the data set before it receives the output. This is the default.

MOD
Append the output to the data set

SHR
Clear the data set before it receives the output. SHR allows multiple allocations of the same data set.

DSDISP requires that DSNAME also be specified.

LSSTYPE(CKD | FB)
Specifies the type of devices within the LSS.

CKD
System z-attached devices. This is the default.

FB Open devices

LSSNBR(*lssnbr***)**
Specifies the LSS number for the action specified with ACTION. Valid values for *lssnbr* are hex characters in the range 00-FF. LSSNBR is required when LSSTYPE FB is specified.

DVCNBR(*dvcnbr***)**
Specifies the device's CCA or LUN for the action specified with ACTION. It is required when LSSTYPE FB is specified.

SNBR(*snbr***)**
Specifies the hexadecimal session number. It is required when the value for ACTION is STAT4ALSS, STAT4ESS, STAT4BLSS, STAT4BESS, STAT4ACGRP, DVCSTAT, GMPSTAT and GMLSTAT.

WAITTIME(*waittime* **| 0)**
Specifies the number of seconds, that SDM must wait for a request to complete. A value of zero specifies that the request should not be timed. Valid values are 0 through 65535. The default is 0.

SUBCHSET(*subchset* **| 0)**
Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in

the Hardware Configuration Dialog (HCD). The valid values are determined by what is supported by the host system's processor and configured for the device.

If you omit the parameter, the command is issued to the device that is logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

RSESSION (control a Global Mirror session)

Use request type RSESSION to control a Global Mirror session. This invokes ANTRQST ILK=ESSRVCS REQUEST=RSESSION.

SNBR(*snbr*)
Specifies the hexadecimal session number. It is required.

VOLSER(*volser*)
Specifies the volume serial number of the device to be used for I/O. VOLSER is mutually exclusive with DEVN. Either VOLSER or DEVN is required.

DEVN(*devn*)
Specifies the hexadecimal device number to be used for I/O. DEVN is mutually exclusive with VOLSER. Either VOLSER or DEVN is required.

ACTION(*action*)
Specifies the action to be taken on the request. It is required. The values for *action* are:

DEFINE
The new Global Mirror physical session will be defined using the value specified with SNBR

UNDEFINE
The Global Mirror physical session specified with SNBR will be removed

START
The Global Mirror logical session is to start forming consistency groups

STOP The Global Mirror logical session is to be terminated

PAUSE
The Global Mirror logical session is to pause forming consistency groups

RESUME
The Global Mirror logical session is to resume forming consistency groups.

If the Global Mirror session was paused in a consistent manner using CGPAUSE, RESUME automatically unsuspends and resyncs the Global Copy pairs in the session.

CGPAUSE
The Global Mirror session it so be paused and the Global Copy pairs in the session suspended so that the secondaries are consistent.

LSSTYPE(CKD | FB)
Specifies the type of devices within the LSS.

CKD
System z-attached devices. This is the default.

FB Open devices

LSSNBR(*lssnbr*)
Specifies the number of the LSS upon which the action, specified with ACTION, will be performed. This is required when a value of DEFINE or UNDEFINE is specified with ACTION.

ESSSERIAL(*esserial*)
Specifies the serial number of the storage control upon which the action, specified with ACTION, will be performed. This is required when a value of DEFINE or UNDEFINE is specified with ACTION.

MSSERIAL(*msserial*)
Specifies the serial number of the storage control of the Master. This is required when a value of START, RESUME, PAUSE or STOP is specified with ACTION.

SBINFO(*sbinfo*)
Specifies the information that describes the subordinate storage controls in a Global Mirror session. This is used with a value of START, RESUME, PAUSE or STOP for ACTION. It is specified once for each subordinate storage control in the session.

The format of *sbinfo* is *mssid*,*sssid*,*serialno* where:

mssid
Is the four-digit SSID number in the master storage control where the control path to this subordinate originates

sssid
Is the four-digit SSID number in the subordinate storage control where the control path from the master storage control terminates

serialno
Is the twelve-digit serial number of the subordinate storage control

The SSID values in a set come from the corresponding values that were used for the PESTPATH command connecting the master storage control and storage control subordinates.

Supply the value for SBINFO in SUPERZAP format, using a comma between every 4 digits. For example, to add a subordinate with a master session SSID of 1111, a subordinate serial number of 2222 and a subordinate serial number of 000000012345, specify:

```
SBINFO(1111,2222,0000,0001,2345)
```

To specify multiple subordinates, use a separate keyword for each subordinate:

```
SBINFO(1111,2222,0000,0001,2345)
SBINFO(3333,4444,0000,0006,7890)
```

MASTER(YES | NO)
Specifies whether the command is issued to the master storage control or a subordinate storage control in the Global Mirror session.

YES
It is issued to the master storage control. This is the default.

NO It is issued to the subordinate storage control, pointed to by VOLSER or DEVN.

FORCE(NO | YES)
Used when STOP is specified with ACTION. Specifies whether the STOP action will wait until a consistent copy of the data has been formed.

NO STOP will not complete until a consistent copy of the data has been formed. This is the default.

YES
STOP will be done even if a consistent copy of the data has not been formed

CGINTERVAL(*cginterval***)**
Specifies, in hexadecimal, the number of seconds between formation of consistency groups. A value of zero means consistency groups will be formed consistently. CGINTERVAL is required if CGDRAIN is specified.

The default is 0.

CGDRAIN(*cgdrain***)**
Specifies, in hexadecimal, the number of seconds allowed for updated tracks to be transferred from the primary volumes to the secondary volumes. The default is the larger of four minutes or twice the time specified with CGINTERVAL. If you specify 0, the value will be set to the default. CGDRAIN requires CGDRAIN and COORDINTERVAL.

COORDINTERVAL(*coordinterval***)**
Specifies, in hexadecimal, the number of milliseconds allotted to allow the master storage control to communicate with subordinates in forming consistency groups. A value of zero requests the default, X'32' (decimal 50). COORDINTERVAL requires CGDRAIN and CGINTERVAL.

WAITTIME(*waittime* **|** **0)**
Specifies the number of seconds that SDM must wait for a request to complete. A value of zero specifies that the request should not be timed. Valid values are 0 through 65535. The default is 0.

SUBCHSET(*subchset* **|** **0)**
Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is supported by the host system's processor and configured for the device.

If you omit the parameter, the command is issued to the device that is logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

RVOLUME (Global Mirror manage volumes)

Use request type RVOLUME to manage Global Mirror volumes. This invokes ANTRQST ILK=ESSRVCS REQUEST=RVOLUME.

SNBR(*snbr***)**
Specifies the hexadecimal Global Mirror session number. It is required.

VOLSER(*volser***)**
Specifies the volume serial number of the device to be used for I/O. VOLSER is mutually exclusive with DEVN. Either VOLSER or DEVN is required.

DEVN(*devn***)**
Specifies the hexadecimal device number of the device to be used for I/O. DEVN is mutually exclusive with VOLSER. Either DEVN or VOLSER is required.

ACTION(JOIN | REMOVE)
Specifies the action to be taken. It is required.

JOIN
Devices specified should be added to the Global Mirror session that is defined with the SNBR parameter

REMOVE
Devices specified should be removed from the Global Mirror session that is defined with the SNBR parameter

LSSNBR(*lssnbr*)
Specifies the number of the LSS upon which the action will be performed. It is required.

ESSSERIAL(*essserial*)
Specifies the serial number of the storage control upon which the action will be performed. It is required.

VOLLIST(*vol1*,*vol2*,....)
Specifies a list of Channel Connection Addresses (CCAs) or Logical Unit Numbers (LUNs) that are affected by the action specified with ACTION. VOLLIST is mutually exclusive with VOLRANGE, MTVOLLIST and MTVOLRANGE. VOLLIST, VOLRANGE, MTVOLLIST or MTVOLRANGE is required. You can include up to 14 2-digit CCAs, separated by commas, with each VOLLIST keyword, and you can specify VOLLIST multiple times. For example, to specify devices from CCA 00 to CCA 07, you could use:

```
VOLLIST(00,01,02,04,05,06,07)
```

or

```
VOLLIST(00,01,02,03)
VOLLIST(04,05,06,07)
```

You can specify up to 26 CCAs with an RVOLUME command.

VOLRANGE(*volrange1*,*xvolrange2*,....)
Specifies a list of ranges of Channel Connection Addresses (CCAs) or Logical Unit Numbers (LUNs) that are affected by the action specified with ACTION. VOLRANGE is mutually exclusive with VOLLIST, MTVOLLIST and MTVOLRANGE. VOLLIST, VOLRANGE, MTVOLLIST or MTVOLRANGE is required. You can specify up to 8 4-digit volume ranges, separated by commas, with each VOLRANGE keyword, and you can specify VOLRANGE multiple times. For example, to specify devices from CCA 00 to CCA 07 and CCA 09 to CCA 14, you could use:

```
VOLRANGE(0007,0914)
```

or

```
VOLRANGE(0007)
VOLRANGE(0914)
```

You can specify up to 31 ranges with an RVOLUME command.

MTVOLLIST(*pcca*,*secser#*,*slss*,*scca*)
Specifies a list of Global Copy device numbers that are affected by the action specified with ACTION.

pcca Specifies the primary CCA (or LUN)

secser# Specifies the serial number of the secondary device's SFI for the Global Copy pair *slss* Specifies the LSS number of the secondary device for the Global Copy pair

scca Specifies the CCA (or LUN) number of the secondary device for the Global Copy pair

You can use the MTVOLLIST keyword up to 32 times to specify more devices. MTVOLLIST is mutually exclusive with VOLLIST, VOLRANGE, and MTVOLRANGE.

MTVOLRANGE(*pcca1:pccax,secser#,slss,scca1*)
Specifies a list of primary device ranges in an LSS that are affected by the action specified with ACTION.

pcca1:pccax
Specifies the first and last primary CCA (or LUN) in a range of contiguous primary devices. The first and last CCA indicated are separated by a colon.

secser# Specifies the serial number of the secondary device's SFI for the Global Copy pair

slss Specifies the LSS number of the secondary device for the Global Copy pair

scca1 Specifies the first CCA (or LUN) of the secondary device for the Global Copy pair. The last secondary CCA is derived from the number of devices specified in the *pcca1:pccax* range.

Up to 32 device set ranges may be specified, specifying the MTVOLRANGE keyword for device set. MTVOLRANGE is mutually exclusive with VOLLIST, VOLRANGE, and MTVOLLIST.

LSSTYPE(CKD | FB)
Specifies the type of devices within the LSS.

CKD
System z-attached devices. This is the default.

FB Open devices

WAITTIME(*waittime* | 0)
Specifies the number of seconds that SDM must wait for a request to complete. A value of zero specifies that the request should not be timed. Valid values are 0 through 65535. The default is 0.

SUBCHSET(*subchset* | 0)
Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is supported by the host system's processor and configured for the device.

If you omit the parameter, the command is issued to the device that is logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

STATESAVE (Request State Save)

Use request type STATESAVE to request a standard or none-disruptive state save, for the storage facility image (SFI) in which a specified device resides, for certain error conditions. This invokes ANTRQST ILK=ESSRVCS REQUEST=STATESAVE.

One statesave is allowed every 5 minutes. However, only 10 non-disruptive statesaves are allowed in a 24-hour period.

DEVN(*devno***)**
Specifies the 2-byte (4 hexadecimal digits) number of the device to which the STATESAVE command is to be sent.

Because a state save is applicable to an entire SFI, the device to which the command is issued does not have to be the device on which an error occurred. IBM recommends that an alternate device in the same SFI be used as the target for the state save, as the device in error may not be available.

NDSS(*statesave***)**
Specifies whether a non-disruptive state save should be taken.

YES
Requests a non-disruptive state save. This does not cause a warm start of the SFI. This is the default.

NO Requests a standard state save, with a resulting warm start in the SFI.

SUBCHSET(*subchset* **|** **0)**
Specifies the subchannel set in which the command is to be issued. The subchannel set where the I/O will be issued is the subchannel set as defined in the Hardware Configuration Dialog (HCD). The valid values are determined by what is supported by the host system's processor and configured for the device.

If you omit the parameter, the command is issued to the device that is logically in subchannel set 0. If a swap has occurred, this could be the device that was defined in an alternate subchannel set in the I/O configuration (IODF), with HCD.

Use the following optional parameters to describe the state save. The values are not checked or validated.

CALLER(*caller***)**
Specifies the name of the caller requesting the state save. The default is `INTERNAL` (the internal caller).

CCA(*cca***)**
Specifies the CCA associated with the device where the initial error occurred.

DIAGREAS(*diagreason***)**
Specifies a diagnostic reason code to be associated with the state save. This can be used to identify the problem that prompted the state save.

DIAGRETC(*diagretc***)**
Specifies diagnostic return code to be associated with the state save. This can be used to identify the problem that prompted the state save.

FUNC(*function***)**
Specifies the function being performed at the time the state save was requested. This is a free-form field that is not checked or validated. You can specify anything that may help explain the conditions at the time that the state save was requested.

LSS(*lss*)
Specifies the the LSS associated with the device where the initial error occurred.

SEQNO(*seqno*)
Specifies the the XRC read record set sequence number associated with the initial error.

SESSION(*session*)
Specifies the XRC session associated with the state save.

TIME(*timestamp*)
Specifies the time, in STCK format, associated with the state save. The default is the time at which the state save I/O request is issued.

TITLE(*title*)
Specifies a title for the state save. This is a free-form text field that is not checked or validated. You can specify anything that helps explain the conditions at the time that the state save was requested. The default is no title text.

TYPE(*functiontype*)
Specifies the type of function that was in control at the time the state save was requested. The values for *functiontype* are:

CC Concurrent Copy

FC FlashCopy

FREEZE
Freeze

GC Global Copy

GM Global Mirror

MM Metro Mirror

PATHS
Establish or delete paths

RUN
Run

SS SnapShot

XRC
Extended Remote Copy

XRCER
XRC Enhanced Reader

WAIT (perform a wait)

Use request type WAIT to perform a wait. This results in the invocation of STIMER.

WAITTIME(*waittime*)
Specifies the number of seconds, *time*, that the program should wait. This results in an STIMER request. *time* is a number between 1 and 1800.

XADD (XRC add volume pair)

Use request type XADD to add XRC volume pairs. This invokes ANTRQST ILK=XRC REQUEST=XADD.

SID(*session*)
Specifies the name of the XRC session. It is the 1 to 8 character name of a session previously specified on the XSTART command. The session name cannot be ALL, as this is reserved, or include imbedded blanks.

PVOLSER(*volser*)
Specifies the MVS volume serial number for the primary volume to be added. PVOLSER is mutually exclusive with VOLLIST and SUSPENDED(YES). Either PVOLSER, VOLLIST or SUSPENDED is required.

SVOLSER(*volser*)
Specifies the MVS volume serial number for the secondary volume to be added. SVOLSER is required if PVOLSER is specified. SVOLSER is mutually exclusive with VOLLIST and SUSPENDED(YES).

VOLLIST(*volser*,*volser*,*volser*...)
Specifies a list of 2 to 100 MVS volume serial numbers of volume pairs to be added. The number of volume serial numbers specified must be an even number. VOLLIST is mutually exclusive with PVOLSER, SVOLSER and SUSPENDED(YES). Either PVOLSER, VOLLIST or SUSPENDED is required.

You can include up to 3 volume pairs, separated by commas, with each VOLLIST keyword, and you can specify VOLLIST multiple times. For example, to specify volume serial XUTIL1 as a utility volume and PAN001 and SAN001 as a volume pair, you could use:

```
VOLLIST(XUTIL1,XRCUTL,PAN001,SAN001)
```

or

```
VOLLIST(XUTIL1,XRCUTL)
VOLLIST(PAN001,SAN001)
```

SUSPENDED(NO | YES)
Indicates whether all suspended volume pairs will be added back into the session.

NO Suspended volume pairs should not be re-added. This is the default.

YES
Suspended volume pairs should be re-added.

SUSPENDED(YES) is mutually exclusive with PVOLSER, SVOLSER, and VOLLIST. Either PVOLSER, VOLLIST or SUSPENDED is required.

COPY(FUL | NO | QIK)
Specifies the type of volume initialization the system should use to copy the primary volume to the secondary volume.

FUL
Full initial copy. This is the default.

NO No initial copy

QIK
Quick initial copy

ERRLVL(*errlvl*)
Specifies the error level that is associated with the volume pair. If this keyword is not specified, the default of SYSTEM is used.

DONOTBLOCK(NO | YES)
Specifies if device blocking should be enabled or disabled.

Workload-based write pacing affects the behavior of DONOTBLOCK. For more information, refer to "Workload-based write pacing" on page 161.

NO Device blocking is enabled. This is the default if DVCBLOCK is not specified.

YES
Device blocking is disabled. DONOTBLOCK(YES) is mutually exclusive with DVCBLOCK.

DVCBLOCK(ON | OFF | WP*n*)
Specifies if device blocking should be enabled or disabled.

Workload-based write pacing affects the behavior of DVCBLOCK. For more information, refer to "Workload-based write pacing" on page 161.

ON Device blocking is enabled.

When workload-based write pacing is in use, XRC automatically converts DVCBLOCK(ON) to DVCBLOCK(WP*n*), where *n* is the discretionary level, 6.

OFF
Doth device blocking and write pacing are disabled

WP*n*
Write pacing is to be activated for the specified volume(s), with *n* specifying the level of write pacing delay, 0-F. WP0 specifies that the session default level will be used, as specified in the SHADOW DfltWritePacingLvl PARMLIB value. For values ranging from WP1-WPF, refer to the ANTRQST macro.

DVCBLOCK is mutually exclusive with DONOTBLOCK(YES).

If neither DONOTBLOCK nor DVCBLOCK is specified, DONOTBLOCK(NO) is the default.

SCSESSION(– – | *scsession*)
Contains an alphabetic value that allows specifying a storage control session name for volume pair. The default is —.

MESSAGES(*msg*)
Specifies a variable, *msg*, in which ANTRQST will place the address of a buffer containing messages about the results of the processed request. These are messages that would have been returned to the caller if the TSO interface had been used. MESSAGES is required. Refer to the ANTRQST macro for the format of the buffer.

WAITTIME(*waittime* | 0)
Specifies the number of seconds that SDM must wait for a request to complete. A value of zero specifies that the request should not be timed. Valid values are 0 through 65535.

XADVANCE (XRC recover)

Use request type XADVANCE to recover an XRC session. This invokes ANTRQST ILK=XRC REQUEST=XADVANCE.

SID(*session*)
Specifies the name of the XRC session to which the request is sent. It is required.

HLQ(*hlqr*)
Specifies the high-level qualifier that was used on the XSTART command.

MHLQ(*mhlqr*)
Specifies the high-level qualifier for the master session control data set that was used on the XCOUPLE command.

ETYPE(WAIT | NOWAIT)
Indicates whether or not ANTRQST will return to the caller before advance processing has completely finished.

WAIT
ANTRQST will not return until advance processing has finished. This is the default.

YES
ANTRQST will return when the advance processing has been scheduled.

MESSAGES(*msg*)
Specifies a variable, *msg*, in which ANTRQST will place the address of a buffer containing messages about the results of the processed request. These are messages that would have been returned to the caller if the TSO interface had been used. Refer to the ANTRQST macro for the format of the buffer. MESSAGES is required.

WAITTIME(*waittime* | 0)
Specifies the number of seconds that SDM must wait for a request to complete. A value of zero specifies that the request should not be timed. Valid values are 0 through 65535. The default is 0.

XCONTIME (XRC consistency time)

Use request type XCONTIME to ask for the current XRC consistency time. This invokes ANTRQST ILK=XRC REQUEST=XCONTIME.

SID(*session*)
Specifies the name of the XRC session to which the request is sent. It is required.

TIME(*times*)
Specifies the name of the field to be used to return consistency time for the specified XRC session. It is required. The returned time is a 64-bit time-of-day (TOD) value in UTC format. The consistency time returned is the time to which volume pairs in the XRC session were last consistent. The time is taken from the application system clock.

HLQ(*hlqr*)
Specifies the high-level qualifier that was used on the XSTART command.

The HLQ name will only be used if XRC is not active. In this case, the consistency time will be obtained from the XRC state data set with the specified HLQ.

WAITTIME(*waittime* | 0)
Specifies the number of seconds that SDM must wait for a request to complete. A value of zero specifies that the request should not be timed. Valid values are 0 through 65535.

XCOUPLE (couple XRC connections)

Use request type XCOUPLE to manage XRC connections. This invokes ANTRQST ILK=XRC REQUEST=XCOUPLE.

SID(*session***)**
Specifies the name of the XRC session to which the request is sent. It is required.

ETYPE(ADD | DELETE | PURGE | RELEASE)
Specifies the type of coupling operation to be processed. It is required.

ADD
The logical session specified should be added to the master session

DELETE
The logical session specified should be deleted from the master session

PURGE
The logical session should be purged (cleaned up) from the master session

RELEASE
A HOLD on the master session should be released

HLQ(*hlqr***)**
Specifies the high-level qualifier of the XRC journal data set.

MHLQ(*mhlqr***)**
Specifies the high-level qualifier of the master control data set. The default is SYS1.

MID(*mid***)**
Specifies the XRC master session ID for the request.

MESSAGES(*msg***)**
Specifies a variable in which ANTRQST will place the address of a buffer containing messages about the results of the processed request. These are messages that would have been returned to the caller if the TSO interface had been used. Refer to the ANTRQST macro for the format of the buffer. This is required.

WAITTIME(*waittime* **| 0)**
Specifies the number of seconds that SDM must wait for a request to complete. A value of zero specifies that the request should not be timed. Valid values are 0 through 65535. The default is 0.

XDEL (XRC delete volume pair)

Use request type XDEL to delete pairs of volumes from an XRC session. This invokes ANTRQST ILK=XRC REQUEST=XDEL.

SID(*session***)**
Specifies the name of the XRC session to which the request is sent. It is required.

PVOLSER(*volser***)**
Specifies the MVS volume serial number for the primary volume to be deleted, or ALL if all of the volume pairs are to be deleted. It is required.

VOLLIST(*volser***,***volser***,***volser***....)**
Specifies a list of 1 to 100 MVS primary volume serial numbers of volume pairs to be deleted. VOLLIST is mutually exclusive with PVOLSER.

You can include up to 6 primary volumes, separated by commas, with each VOLLIST keyword, and you can specify VOLLIST multiple times.

ATTIME(*timestamp***)**
Specifies the UTC timestamp of when the volumes are to be deleted. The

format is YYYY.DDD_HH:MM:SS.THMIJU. This field is required if the ETYPE parameter ATTIME is specified, otherwise, it is ignored.

ETYPE(IMMEDIATE | ATTIME | CANCEL | DRAIN)
Specifies when the command will be processed.

IMMEDIATE
Process the XDEL command as soon as the current consistency group has been applied. This is the default.

ATTIME
Process the XDEL command when the specified UTC time is reached

CANCEL
Delete an outstanding XDEL command

DRAIN
Process the XDEL command when the consistency group represented by the most recent timestamp of all volumes being deleted has been applied

MESSAGES(*msg*)
Specifies a variable, *msg*, in which ANTRQST will place the address of a buffer containing messages about the results of the processed request. These are messages that would have been returned to the caller if the TSO interface had been used. Refer to the ANTRQST macro for the format of the buffer. MESSAGES is required.

WAITTIME(*waittime* | 0)
Specifies the number of seconds that SDM must wait for a request to complete. A value of zero specifies that the request should not be timed. Valid values are 0 through 65535. The default is 0.

XEND (XRC end)

Use request type XEND to end an XRC session. This invokes ANTRQST ILK=XRC REQUEST=XEND.

SID(*session*)
Specifies the name of the XRC session to which the request is sent. It is required.

ATTIME(*timestamp*)
Specifies the UTC timestamp for when the volumes are to be deleted. The format is YYYY.DDD_HH:MM:SS.THMIJU. This field is required if ETYPE(ATTIME) is specified; otherwise, it is ignored.

ETYPE(IMMEDIATE | ATTIME | CANCEL | DRAIN)
Specifies when the command will be processed.

IMMEDIATE
Process the XEND command as soon as the current consistency group has been applied. This is the default.

ATTIME
Process the XEND command when the specified UTC time is reached

CANCEL
Delete an outstanding XEND command

DRAIN
Process the XEND command when the consistency group represented by the most recent timestamp of all volumes being deleted has been applied

MESSAGES(*msg***)**
Specifies a variable in which ANTRQST will place the address of a buffer containing messages about the results of the processed request. These are messages that would have been returned to the caller if the TSO interface had been used. Refer to the ANTRQST macro for the format of the buffer. This is required.

WAITTIME(*waittime* **| 0)**
Specifies the number of seconds that SDM must wait for a request to complete. A value of zero specifies that the request should not be timed. Valid values are 0 through 65535. The default is 0.

XQUERY (XRC query)

Use request type XQUERY to query an XRC session. This invokes ANTRQST ILK=XRC REQUEST=XQUERY.

SID(*session***)**
Specifies the name of the XRC session to which the request is sent. It is required.

DSNAME(*dataset-name***)**
Specifies that XRC direct the XQUERY output to a sequential data set that is identified by *dataset-name*. It must be large enough to accommodate the requested XQUERY reports. DSNAME is mutually exclusive with MESSAGES.

DSDISP(OLD | MOD | SHR)
Specifies how XRC writes the XQUERY output to the data set:

OLD
Clear the data set before it receives the output. This is the default.

MOD
Append the output to the data set

SHR
Clear the data set before it receives the output. SHR allows multiple allocations of the same data set.

DSDISP requires the DSNAME parameter.

ETYPE(SUMMARY | *etype***)**
Specifies which query command to execute.

CONFIGURATION
XQUERY CONFIGURATION report

ENVIRONMENTF
XQUERY ENVIRONMENT(FLAG) report

ENVIRONMENTH
XQUERY ENVIRONMENT(PATCH) report

ENVIRONMENTP
XQUERY ENVIRONMENT(PARMLIB) report

DETAIL
XQUERY VOLUME_DETAIL report

DETAILS
XQUERY STORAGECONTROL DETAIL report

MASTER
XQUERY MASTER report

PACE
XQUERY PACE report

SET
XQUERY SET report

SWAP
XQUERY SWAP report

STORAGECONTROL
XQUERY STORAGECONTROL report

FEATURES
XQUERY STORAGECONTROL FEATURES report

SUMMARY
XQUERY SUMMARY report

VOLUME
XQUERY VOLUME report

XFEATURES
XQUERY STORAGECONTROL XFEATURES report

MHLQ(*mhlqr* **)**
Contains the high-level qualifier of the master control data set. This field is used only if ETYPE(MASTER) is specified. The default is SYS1.

SCSESSION(– – | *scsession***)**
Contains an alphabetic value that allows requesting that only volumes in the specified storage control session be displayed. This field is ignored for all values for the ETYPE parameter except DETAIL and VOLUME. The default is —.

SSID(*ssid***)**
Contains a 2-byte hexadecimal number that allows only volumes in the specified storage subsystem be displayed. This field is ignored for all values for the ETYPE parameter except DETAIL and VOLUME.

VOLLIST(*volser***,***volser***,***volser***....)**
Specifies a list of 1 to 100 MVS primary volume serial numbers of volume pairs. You can include up to 6 primary volumes, separated by commas, with each VOLLIST keyword, and you can specify VOLLIST multiple times.

MESSAGES(*msg***)**
Specifies a variable in which ANTRQST will place the address of a buffer containing messages about the results of the processed request. These are messages that would have been returned to the caller if the TSO interface had been used. Refer to the ANTRQST macro for the format of the buffer. MESSAGES is mutually exclusive with DSNAME. This is required.

WAITTIME(*waittime* **| 0)**
Contains a number of seconds that SDM must wait for a request to complete. A value of zero specifies that the request should not be timed. Valid values are 0 through 65535. The default is 0.

XRECOVER (XRC recover)

Use request type XQUERY to recover an XRC session. This invokes ANTRQST ILK=XRC REQUEST=XQUERY.

SID(*session***)**
Specifies the name of the XRC session. It is required.

CHECK(YES|NO)
Specifies if **only** XRECOVER enhanced checking should be performed, bypassing other XRECOVER processing. This produces an XRECOVER CHECK report.

NO Do not perform XRECOVER enhanced checking. This is the default.

YES
Perform only XRECOVER enhanced checking, bypassing other XRECOVER processing. This produces an XRECOVER CHECK report.

CHECK(YES) is mutually exclusive with FORCE(YES).

ETYPE(WAIT | NOWAIT)
Specifies whether or not ANTRQST will return to the caller before recovery has completely finished.

WAIT
Indicates ANTRQST will not return until recovery processing has finished. This is the default.

NOWAIT
Indicates ANTRQST will return when recovery processing has been scheduled.

FORCE(YES|NO)
Specifies whether or not to perform XRECOVER enhanced checking.

NO Indicates that XRECOVER enhanced checking should be performed. This is the default.

YES
Indicates XRECOVER enhanced checking should not be performed.

FORCE(YES) is mutually exclusive with CHECK(YES).

HLQ(*hlqr*)
Contains the high-level qualifier that was used on the XSTART command. The default is SYS1.

MHLQ(*mhlqr*)
Contains the high-level qualifier that was used on the XCOUPLE command for the master control data set. The default is SYS1.

ONLINE(NO | YES)
Contains a keyword indicating what type of test to use to determine if a secondary volume can be used as a recovery volume.

NO Only the volume added as a secondary can be used as a recovery volume. This is the default.

YES
Any volume that is online and has a volume serial number matching the desired secondary volume can be used as a recovery volume

TERTIARY(NO | YES)
Contains a keyword indicating what type of test to use to determine if a secondary volume can be used as a recovery volume.

NO Only the volume added as a secondary can be used as a recovery volume. This is the default.

YES
Any volume that is online and has a volume serial number matching the desired secondary volume can be used as a recovery volume, except for the original secondary volume itself

MESSAGES(*msg*)
Specifies a variable in which ANTRQST will place the address of a buffer containing messages about the results of the processed request. These are messages that would have been returned to the caller if the TSO interface had been used. Refer to the ANTRQST macro for the format of the buffer. This is required.

WAITTIME(*waittime* | 0)
Contains a number of seconds that SDM must wait for a request to complete. A value of zero specifies that the request should not be timed. Valid values are 0 through 65535. The default is 0.

XSCSTATS (XRC statistics)

Use request type XSCSTATS to provide statistics for storage control System Data Mover sessions known to the system this command is processed on. This invokes ANTRQST ILK=XRC REQUEST=XSCSTATS.

ETYPE(SUMMARY | SINGLE)
Specifies what type of statistics to be returned.

SUMMARY
A summary of data mover sessions (scsessions) maintained in the storage controls attached to the system should be returned. This is the default.

SINGLE
A complete set of statistics for a specified data mover session (scsessions) should be returned. SSID and SCSESSION must be specified to allow locating the session.

SCSESSION(– – | *scsession*)
Specifies the storage control identifier (SCID on the XQUERY VOLUME_DETAIL_REPORT) to be used for gathering statistics. The default is – –.

SSID(*ssid*)
Specifies what storage control will be used to gather statistics.

QRYINFO(*qryinfo*)
Creates the data area where the query information is to be returned. Either QRYINFO or DSPINFO is required with XSCSTATS. This area is the size specified in the QRYSIZE keyword. QRYINFO is mutually exclusive with DSPINFO.

QRYSIZE(*qrysize*)
Specifies the decimal length of the QRYINFO area. It is required.

DSPINFO(*dspinfo*)
Names a variable output field to be used to return the result of the display. Either QRYINFO or DSPINFO is required with XSCSTATS. DSPINFO is mutually exclusive with QRYINFO.

WAITTIME(*waittime* | 0)
Contains a number of seconds that SDM must wait for a request to complete. A value of zero specifies that the request should not be timed. Valid values are 0 through 65535. The default is 0.

XSET (XRC session parameters)

Use request type XSET to set XRC session parameters. This invokes ANTRQST ILK=XRC REQUEST=XSET.

SID(*session*)
Specifies the name of the XRC session to which the request is sent. It is required.

DVCBLOCK(ON | WP*n* | OFF)
Specifies whether device blocking or write pacing is enabled or disabled.

Workload-based write pacing affects the behavior of DVCBLOCK. For more information, refer to "Workload-based write pacing" on page 161.

ON Device blocking is enabled. This is the default.

When workload-based write pacing is in use, XRC automatically converts DVCBLOCK(ON) to DVCBLOCK(WP*n*), where *n* is the discretionary level, 6.

OFF
Both device blocking and write pacing are disabled

WP*n*
Specifies that write pacing is to be activated for the specified volume(s), with *n* specifying the level of write pacing delay, 0-F. WP0 specifies that the session default level will be used, as specified in the SHADOW DfltWritePacingLvl PARMLIB value. For values ranging from WP1-WPF, please refer to the ANTRQST macro.

PVOLSER(ALLB | ALL | ALLDP | ALLP | *volser*)
When specifying the DVCBLOCK, this optional input field specifies the name of the primary volume serial of a volume that is to have its device blocking or write pacing status changed.

ALLB
All primary volumes set to DVCBLOCK=ON are to be changed. This is the default.

ALL
All primary volumes are to be changed

ALLDP
All primary volumes set to write pacing level WP0 are to be changed

ALLP
All primary volumes set to write pacing level WP1-WPF are to be changed

volser
MVS volume serial number for the volume to be processed according to the value specified in the DVCBLOCK parameter

REFRESHP(NO | YES)
Specifies whether the system data mover should examine the capabilities of the specified primary storage control(s) and update the data mover's internal control information. REFRESHP is mutually exclusive with SUSLBUSY and TIMEOUT.

NO System data mover control information remains unchanged. This is the default.

YES
System data mover should examine the capabilities and update the data mover's internal control information

SSID may be specified with REFRESHP to identify the primary storage control(s) to be examined. If it is omitted, all primary storage controls associated will be examined.

COPY(FUL | QIK)
Specifies what type of volume initialization should be used to copy a primary volume to a secondary volume.

FUL
Full initial copy. This is the default.

QIK
Quick initial copy

PAGEFIX(8 | *pagefix***)**
Specifies the number of megabytes assigned to the data mover as permanently page-fixed real storage. 8MB is the default.

PMEMBER(*parmlib-member***)**
Contains a PARMLIB member name.

PDSNAME(*pds-name***)**
Contains the name of a partitioned data set that contains PMEMBER (up to 44 characters).

PACTION(V | A)
Indicates whether the PARMLIB member is to be verified or verified and then applied.

V Verify the PARMLIB member. This is the default.

A Verify the PARMLIB member, and, if no errors are found, apply it

PRIORITY(LOAD | FIFO)
Indicates how the next volume to be synchronized or resynchronized will be selected.

LOAD
the next volume is based on which controller has the least load. This is the default.

FIFO
The next volume is based on the order of the XADDPAIR commands

RFREQUENCY(00.00.00 | *hh.mm.ss***)**
Specifies how often the software bitmap is toggled. The format is HH.MM.SS. 00:00:00 specifies that time-controlled toggling is deactivated. This is the default.

RTRACKS(7500 | *rtracks***)**
Specifies how many tracks have changed before the software bitmap is toggled. To deactivate track-controlled toggling, specify 0. Valid values are 0–99999. The default is 7500.

SCSYNCHP(2 | *sc-synchp***)**
Specifies the maximum number of concurrent synchronization or resynchronization tasks allowed per primary storage controller. Valid values are 0–45. The default is 2.

SCSYNCHS(*sc-synchs*)
Specifies the maximum number of concurrent synchronization or resynchronization tasks allowed per secondary storage controller. Valid values are 0–45. The default is 2.

SSID(ALL | *ssid*)
Specifies the storage controls that are to be processed by the XSET TIMEOUT, REFRESHP or SUSLBUSY command. ALL indicates all storage controls are to be processed.

SUSLBUSY(NO | YES)
Enables or disables the Suspend on Long Busy function. SUSLBUSY is mutually exclusive with REFRESHP and TIMEOUT.

NO Suspend on Long Busy is disabled. This is the default.

YES
Suspend on Long Busy is enabled. If associated SSID(s) are specified, the change will be made to just those SSID(s). If they are not specified then this will become the storage default.

SYNCH(4 | *synch*)
Specifies the maximum number of concurrent synchronization or resynchronization tasks allowed per session. Valid values are 0–45. The default is 4.

TIMEOUT(*hh.mm.ss*)
Specifies the primary control timeout value. The format is HH.MM.SS. The storage control default is usually 00.05.00. The value can range from a minimum of one second (00.00.01) to a maximum of 18 hours (18.00.00) TIMEOUT is mutually exclusive with SUSLBUSY and REFRESHP.

UTILITY(FLT | FIX)
Indicates what type of utility device support should be enabled when an ADDPAIR is done with a secondary volume of XRCUTL.

FLT
Floating utility device support. This is the default.

FIX
Fixed utility device support

MESSAGES(*msg*)
Specifies a variable in which ANTRQST will place the address of a buffer containing messages about the results of the processed request. These are messages that would have been returned to the caller if the TSO interface had been used. Refer to the ANTRQST macro for the format of the buffer.

WAITTIME(*waittime* | 0)
Contains a number of seconds that SDM must wait for a request to complete. A value of zero specifies that the request should not be timed. Valid values are 0 through 65535. The default is 0.

XSTART (XRC start)

Use request type XSTART to start an XRC session. This invokes ANTRQST ILK=XRC REQUEST=XSTART.

SID(*session*)
Specifies the name of the XRC session to which the request is sent. It is required.

STYPE(*stype*)
Contains the type of XRC session identified by SID. It is required.

ERRLVL(*errlvl*)
Contains an error level that is associated with volume pairs added to the session. It is required.

HLQ(*hlqr*)
Contains the high-level qualifier of the state, control and journal data sets. The default is SYS1.

MESSAGES(*msg*)
Specifies a variable in which ANTRQST will place the address of a buffer containing messages about the results of the processed request. MESSAGES is required. These are messages that would have been returned to the caller if the TSO interface had been used. Refer to the ANTRQST macro for the format of the buffer.

WAITTIME(*waittime* | 0)
Contains a number of seconds that SDM must wait for a request to complete. A value of zero specifies that the request should not be timed. Valid values are 0 through 65535. The default is 0.

XSTATUS (XRC status)

Use request type XSTATUS to return status on an XRC session. This invokes ANTRQST ILK=XRC REQUEST=XSTATUS.

ETYPE(ADDRSPACE | CLUSTER)
Specifies the type of the XSTATUS query. It is required.

ADDRSPACE
Display the names of all active XRC address spaces in the LPAR. This is the default.

CLUSTER
Display information about coupled XRC sessions associated with the cluster session in the LPAR.

DSNAME(*dataset-name*)
Specifies that XRC direct the XSTATUS output to a sequential data set that is identified by *dataset-name*. The data set must already be allocated and must be large enough to accommodate the requested XSTATUS reports. DSNAME is mutually exclusive with MESSAGES.

DSDISP(OLD | MOD | SHR)
Specifies how XRC writes the XSTATUS output to the data set:

OLD
Clear the data set before it receives the output. This is the default.

MOD
Append the output to the data set.

SHR
Clear the data set before it receives the output. SHR allows multiple allocations of the same data set.

DSDISP requires DSNAME.

MESSAGES(*msg*)
Specifies a variable in which ANTRQST will place the address of a buffer containing messages about the results of the processed request. These are

messages that would have been returned to the caller if the TSO interface had been used. Refer to the ANTRQST macro for the format of the buffer.

WAITTIME(*waittime* **| 0)**
Contains a number of seconds that SDM must wait for a request to complete. A value of zero specifies that the request should not be timed. Valid values are 0 through 65535. The default is 0.

XSUSPEND (XRC suspend)

Use request type XSUSPEND to suspend an XRC session. This invokes ANTRQST ILK=XRC REQUEST=XSUSPEND.

SID(*session***)**
Specifies the name of the XRC session to which the request is sent. It is required.

PVOLSER(*volser***)**
Specifies the MVS volume serial number for the primary volume to be suspended. PVOLSER is mutually exclusive with TIMEOUT and SCSESSION.

TIMEOUT(00.00.00 | *hh.mm.ss***)**
Specifies a timeout value that will be sent to every controller in the session as part of the session suspension. The format is HH.MM.SS. TIMEOUT is mutually exclusive with PVOLSER, SCSESSION and VOLLIST.

The default is 00.00.00.

VOLLIST(*volser***,***volser***,***volser***....)**
Specifies a list of 1 to 100 MVS primary volume serial numbers of volume pairs to be suspended. VOLLIST is mutually exclusive with PVOLSER, SCSESSION and TIMEOUT. You can include up to 6 primary volumes, separated by commas, with each VOLLIST keyword, and you can specify VOLLIST multiple times.

SCSESSION(*scsession***)**
Specifies a hexadecimal storage control session ID. DEVN is required with SCSESSION. SCSESSION is mutually exclusive with PVOLSER, VOLLIST and TIMEOUT.

DEVN(*devn***)**
Specifies the hexadecimal device number of a device attached to the storage control that will have its session suspended. DEVN is required with SCSESSION.

OPTION(NONE | SYNC)
Specifies how the XSUSPEND VOLUME(ALL) command will be processed:

NONE
Suspensions will take place in parallel. This is the default.

SYNC
Suspensions will take place synchronously

ATTIME(*timestamp***)**
Contains the UTC timestamp of when the suspension will happen. The format is YYYY.DDD_HH:MM:SS.THMIJU. This is required with the ETYPE(ATTIME).

ETYPE(IMMEDIATE | ATTIME | CANCEL | DRAIN)
Specifies when the command will be processed.

IMMEDIATE
Process the XSUSPEND command as soon as the current consistency group has been applied. This is the default.

ATTIME
Process the XSUSPEND command when the specified UTC time is reached

CANCEL
Delete an outstanding XSUSPEND command

DRAIN
Process the XSUSPEND command when the consistency group represented by the most recent timestamp of all volumes being deleted has been applied

MHLQ(*mhlqr*)
Contains the high-level qualifier for the master session control data set.

MESSAGES(*msg*)
Specifies a variable in which ANTRQST will place the address of a buffer containing messages about the results of the processed request. These are messages that would have been returned to the caller if the TSO interface had been used. Refer to the ANTRQST macro for the format of the buffer. MESSAGES is required.

WAITTIME(*waittime* | 0)
Contains a number of seconds, that SDM must wait for a request to complete. A value of zero specifies that the request should not be timed. Valid values are 0 through 65535. The default is 0.

End of Programming Interface Information

Appendix E. Accessibility

Accessible publications for this product are offered through IBM Knowledge Center (http://www.ibm.com/support/knowledgecenter/SSLTBW/welcome).

If you experience difficulty with the accessibility of any z/OS information, send a detailed message to the "Contact us" web page for z/OS (http://www.ibm.com/systems/z/os/zos/webqs.html) or use the following mailing address.

IBM Corporation
Attention: MHVRCFS Reader Comments
Department H6MA, Building 707
2455 South Road
Poughkeepsie, NY 12601-5400
United States

Accessibility features

Accessibility features help users who have physical disabilities such as restricted mobility or limited vision use software products successfully. The accessibility features in z/OS can help users do the following tasks:

- Run assistive technology such as screen readers and screen magnifier software.
- Operate specific or equivalent features by using the keyboard.
- Customize display attributes such as color, contrast, and font size.

Consult assistive technologies

Assistive technology products such as screen readers function with the user interfaces found in z/OS. Consult the product information for the specific assistive technology product that is used to access z/OS interfaces.

Keyboard navigation of the user interface

You can access z/OS user interfaces with TSO/E or ISPF. The following information describes how to use TSO/E and ISPF, including the use of keyboard shortcuts and function keys (PF keys). Each guide includes the default settings for the PF keys.

- *z/OS TSO/E Primer*
- *z/OS TSO/E User's Guide*
- *z/OS V2R2 ISPF User's Guide Vol I*

Dotted decimal syntax diagrams

Syntax diagrams are provided in dotted decimal format for users who access IBM Knowledge Center with a screen reader. In dotted decimal format, each syntax element is written on a separate line. If two or more syntax elements are always present together (or always absent together), they can appear on the same line because they are considered a single compound syntax element.

Each line starts with a dotted decimal number; for example, `3` or `3.1` or `3.1.1`. To hear these numbers correctly, make sure that the screen reader is set to read out punctuation. All the syntax elements that have the same dotted decimal number (for example, all the syntax elements that have the number 3.1) are mutually exclusive alternatives. If you hear the lines 3.1 USERID and 3.1 SYSTEMID, your syntax can include either USERID or SYSTEMID, but not both.

The dotted decimal numbering level denotes the level of nesting. For example, if a syntax element with dotted decimal number 3 is followed by a series of syntax elements with dotted decimal number 3.1, all the syntax elements numbered 3.1 are subordinate to the syntax element numbered 3.

Certain words and symbols are used next to the dotted decimal numbers to add information about the syntax elements. Occasionally, these words and symbols might occur at the beginning of the element itself. For ease of identification, if the word or symbol is a part of the syntax element, it is preceded by the backslash (\) character. The * symbol is placed next to a dotted decimal number to indicate that the syntax element repeats. For example, syntax element *FILE with dotted decimal number 3 is given the format 3 * FILE. Format 3* FILE indicates that syntax element FILE repeats. Format 3* * FILE indicates that syntax element * FILE repeats.

Characters such as commas, which are used to separate a string of syntax elements, are shown in the syntax just before the items they separate. These characters can appear on the same line as each item, or on a separate line with the same dotted decimal number as the relevant items. The line can also show another symbol to provide information about the syntax elements. For example, the lines 5.1*, 5.1 LASTRUN, and 5.1 DELETE mean that if you use more than one of the LASTRUN and DELETE syntax elements, the elements must be separated by a comma. If no separator is given, assume that you use a blank to separate each syntax element.

If a syntax element is preceded by the % symbol, it indicates a reference that is defined elsewhere. The string that follows the % symbol is the name of a syntax fragment rather than a literal. For example, the line 2.1 %OP1 means that you must refer to separate syntax fragment OP1.

The following symbols are used next to the dotted decimal numbers.

? indicates an optional syntax element
The question mark (?) symbol indicates an optional syntax element. A dotted decimal number followed by the question mark symbol (?) indicates that all the syntax elements with a corresponding dotted decimal number, and any subordinate syntax elements, are optional. If there is only one syntax element with a dotted decimal number, the ? symbol is displayed on the same line as the syntax element, (for example 5? NOTIFY). If there is more than one syntax element with a dotted decimal number, the ? symbol is displayed on a line by itself, followed by the syntax elements that are optional. For example, if you hear the lines 5 ?, 5 NOTIFY, and 5 UPDATE, you know that the syntax elements NOTIFY and UPDATE are optional. That is, you can choose one or none of them. The ? symbol is equivalent to a bypass line in a railroad diagram.

! indicates a default syntax element
The exclamation mark (!) symbol indicates a default syntax element. A dotted decimal number followed by the ! symbol and a syntax element indicate that the syntax element is the default option for all syntax elements that share the same dotted decimal number. Only one of the syntax elements that share the dotted decimal number can specify the ! symbol. For example, if you hear the lines 2? FILE, 2.1! (KEEP), and 2.1 (DELETE), you know that (KEEP) is the

default option for the FILE keyword. In the example, if you include the FILE keyword, but do not specify an option, the default option KEEP is applied. A default option also applies to the next higher dotted decimal number. In this example, if the FILE keyword is omitted, the default FILE(KEEP) is used. However, if you hear the lines 2? FILE, 2.1, 2.1.1! (KEEP), and 2.1.1 (DELETE), the default option KEEP applies only to the next higher dotted decimal number, 2.1 (which does not have an associated keyword), and does not apply to 2? FILE. Nothing is used if the keyword FILE is omitted.

*** indicates an optional syntax element that is repeatable**
The asterisk or glyph (*) symbol indicates a syntax element that can be repeated zero or more times. A dotted decimal number followed by the * symbol indicates that this syntax element can be used zero or more times; that is, it is optional and can be repeated. For example, if you hear the line 5.1* data area, you know that you can include one data area, more than one data area, or no data area. If you hear the lines 3* , 3 HOST, 3 STATE, you know that you can include HOST, STATE, both together, or nothing.

Notes:

1. If a dotted decimal number has an asterisk (*) next to it and there is only one item with that dotted decimal number, you can repeat that same item more than once.
2. If a dotted decimal number has an asterisk next to it and several items have that dotted decimal number, you can use more than one item from the list, but you cannot use the items more than once each. In the previous example, you can write HOST STATE, but you cannot write HOST HOST.
3. The * symbol is equivalent to a loopback line in a railroad syntax diagram.

+ indicates a syntax element that must be included
The plus (+) symbol indicates a syntax element that must be included at least once. A dotted decimal number followed by the + symbol indicates that the syntax element must be included one or more times. That is, it must be included at least once and can be repeated. For example, if you hear the line 6.1+ data area, you must include at least one data area. If you hear the lines 2+, 2 HOST, and 2 STATE, you know that you must include HOST, STATE, or both. Similar to the * symbol, the + symbol can repeat a particular item if it is the only item with that dotted decimal number. The + symbol, like the * symbol, is equivalent to a loopback line in a railroad syntax diagram.

Notices

This information was developed for products and services offered in the U.S.A. or elsewhere.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

Site Counsel
IBM Corporation
2455 South Road
Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

COPYRIGHT LICENSE:

This information might contain sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Policy for unsupported hardware

Various z/OS elements, such as DFSMS, HCD, JES2, JES3, and MVS™, contain code that supports specific hardware servers or devices. In some cases, this device-related element support remains in the product even after the hardware devices pass their announced End of Service date. z/OS may continue to service element code; however, it will not provide service related to unsupported hardware devices. Software problems related to these devices will not be accepted for service, and current service activity will cease if a problem is determined to be associated with out-of-support devices. In such cases, fixes will not be issued.

Minimum supported hardware

The minimum supported hardware for z/OS releases identified in z/OS announcements can subsequently change when service for particular servers or devices is withdrawn. Likewise, the levels of other software products supported on a particular release of z/OS are subject to the service support lifecycle of those products. Therefore, z/OS and its product publications (for example, panels, samples, messages, and product documentation) can include references to hardware and software that is no longer supported.

- For information about software support lifecycle, see: IBM Lifecycle Support for z/OS (http://www.ibm.com/software/support/systemsz/lifecycle/)
- For information about currently-supported IBM hardware, contact your IBM representative.

Programming interface information

This publication primarily documents information that is NOT intended to be used as programming interfaces.

This publication also documents an intended programming interface that allows the customer to write programs to obtain the services of the z/OS System Data Mover (SDM). This information is identified where it occurs, either by an introductory statement to a chapter or section or by the following marking:

Programming Interface Information

End of Programming Interface Information

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available at Copyright and Trademark information (http://www.ibm.com/legal/copytrade.shtml).

Glossary

This glossary defines technical terms and abbreviations used in this document. If you do not find the term you are looking for, refer to the index of this document, or view IBM Dictionary of Computing, located at: **www.ibm.com/networking/nsg/nsgmain.htm**.

ANTAS*nnn*
: A generic address space identifier that refers to any one of the following five XRC session address spaces running concurrently in a single LPAR using CXRC or MXRC: ANTAS001, ANTAS002, ANTAS003, ANTAS004, or ANTAS005.

AOM Asynchronous operations manager.

APAR Authorized program analysis report.

asynchronous operation
: A type of operation in which the remote copy XRC or PPRC-XD function copies updates to the secondary volume of a pair at some time after the primary volume is updated. Contrast with *synchronous operation*.

ATTIME
: A keyword for requesting deletion or suspension at a specific target time.

backup
: The process of creating a copy of data to ensure against accidental loss.

cache A random access electronic storage in selected storage controls used to retain frequently used data for faster access by the channel.

CCA Channel connection address.

CCW Channel command word.

CEC Central electronics complex.

channel
: (1) A path along which signals can be sent; for example, data channel and output channel. (2) A functional unit, controlled by the processor, that handles the transfer of data between processor storage and local peripheral equipment.

channel connection address (CCA)
: The input/output (I/O) address that uniquely identifies an I/O device to the channel during an I/O operation.

channel interface
: The circuitry in a storage control that attaches storage paths to a host channel.

CLIST TSO command list.

cluster
: See *storage cluster*.

concurrent copy
: A copy services function that produces a backup copy and allows concurrent access to data during the copy.

consistency group time
: zzThe time, expressed as a primary application system time-of-day (TOD) value, to which secondary volumes have been updated.

control data set
: A data set that contains consistent group information on the secondary volumes and the journal data set. It contains information necessary for recovery operations and acts as the table of contents for the session. The control data set keeps track of data written to secondary volumes, the location of unwritten data in the journal set, and which group to start recovery with.

control unit address (CUA)
: The high order bits of the storage control address, used to identify the storage control to the host system.

: **Note:** The control unit address bits are set to zeros for ESCON attachments.

coupled extended remote copy (CXRC)
: An enhancement to XRC that provides synchronous copy operations in large environments, with an expanded number of primary storage controls and DASD volumes, in excess of those supported by a single system data mover configuration. Installations may have configurations

consisting of thousands of volumes in multiple XRC sessions, with coordination between the sessions to ensure that all volumes can be recovered to a consistent point in time. This greatly expands upon the ability of XRC to provide remote disaster recovery protection across an entire sysplex.

CUA Control unit address.

CXRC Coupled extended remote copy.

DASD
Direct access storage device.

DASD fast write
A form of fast write to cache where the data is written concurrently to cache and nonvolatile storage and automatically scheduled for destaging to the DASD. Both copies are retained in the storage control until the data is completely written to the DASD, providing data integrity equivalent to writing directly to the DASD. DASD fast write is available with cached IBM storage controls.

DASD subsystem
A DASD storage control and its attached direct access storage devices.

data in transit
The update data on application system DASD volumes that is being sent to the recovery system for writing to DASD volumes on the recovery system.

dependent write
An application I/O that depends upon a previous application I/O having completed.

destage
The asynchronous write of new or updated data from cache or nonvolatile storage to DASD. The fast write, dual copy, and remote copy functions destage data. See also *write hit*.

device address
Three or four hexadecimal digits that uniquely define a physical I/O device on a channel path in System/370 mode. The one or two leftmost digits are the address of the channel to which the device is attached. The two rightmost digits represent the unit address.

device ID
An 8-bit identifier that uniquely identifies a physical I/O device.

device number
Four hexadecimal digits that logically identify an open or 390 I/O device.

Device Support Facilities program (ICKDSF)
A program used to initialize DASD at installation and perform media maintenance.

DFDSS
Data Facility Data Set Services.

DFSMSdss
A functional component of DFSMS/MVS used to copy, dump, move, and restore data sets and volumes.

director
See *storage director* and *ESCON Director*.

disaster recovery
Recovery after a disaster, such as a fire, that destroys or otherwise disables a system. Disaster recovery techniques typically involve restoring data to a second (recovery) system, then using the recovery system in place of the destroyed or disabled application system. See also *recovery*, *backup*, and *recovery system*.

DRAIN
A keyword for requesting deletion or suspension when all existing record updates from the storage control cache have been cleared.

dump A capture of valuable storage information at the time of an error.

duplex pair
A volume comprised of two physical devices within the same or different storage subsystems that are defined as a pair by a dual copy, PPRC, or XRC operation, and are in neither suspended nor pending state. The operation records the same data onto each volume.

ECSA Extended common service area.

environmental data
Data that the storage control must report to the host; the data can be service information message (SIM) sense data, logging mode sense data, an error condition that prevents completion of an

asynchronous operation, or a statistical counter overflow. The storage control reports the appropriate condition as unit check status to the host during a channel initiated selection. Sense byte 2, bit 3 (environmental data present) is set to 1.

Environmental Record Editing and Printing (EREP) program
The program that formats and prepares reports from the data contained in the error recording data set (ERDS).

EREP Environmental Record Editing and Printing Program.

ERP Error recovery procedure.

ESCD ESCON Director.

ESCM ESCON Manager.

ESCON
Enterprise System Connection. This is a set of IBM products and services that provides a dynamically connected environment within an enterprise.

ESCON Director (ESCD)
A device that provides connectivity capability and control for attaching any two ESCON links to each other.

ESCON Manager (ESCM)
A licensed program that provides host control and intersystem communication capability for ESCON Director connectivity operations.

ESS Enterprise Storage Server.

extended remote copy (XRC)
A hardware- and software-based remote copy service option that provides an asynchronous volume copy across storage subsystems for disaster recovery, device migration, and workload migration.

failback
An operation that switches back from a redundant or standby system to the primary system when services are restored.

failover
An operation that switches to a redundant or standby system when services fail.

fibre-channel connection
A fibre-channel communications protocol designed for IBM mainframe computers and peripherals.

fibre-channel protocol
A fibre-channel communications protocol that allows access to SCSI devices.

fiber optic cable
A fiber, or bundle of fibers, in a structure built to meet optic, mechanical, and environmental specifications.

FICON
See *fibre-channel connection*.

fixed utility volume
A simplex volume assigned by the storage administrator to a logical storage subsystem to serve as working storage for XRC functions on that storage subsystem.

FlashCopy
A point-in-time copy services function that can quickly copy data from a source location to a target location.

floating utility volume
Any volume of a pool of simplex volumes assigned by the storage administrator to a logical storage subsystem to serve as dynamic storage for XRC functions on that storage subsystem.

GB Gigabyte.

Geographically Dispersed Parallel Sysplex (GDPS)
A multi-site application availability solution that provides the capability to manage remote copy configuration storage subsystems and automate Parallel Sysplex operational tasks. All GDPS functions can be performed from a single point of control, thereby simplifying system resource management. GDPS is designed to minimize and potentially eliminate the impact of any failure or planned site outage.

gigabyte
1 073 741 824 bytes.

global copy for ESS
PPRC-Extended Distance, or PPRC-XD.

global mirror for ESS
: PPRC-Extended Distance, or PPRC-XD combined with FlashCopy consistency groups.

global mirror for System z
: Extended Remote Copy, or XRC.

GTF Generalized trace facility.

ICKDSF
: See *Device Support Facilities program*.

identifier (ID)
: A sequence of bits or characters that identifies a program, device, storage control, or system.

IML Initial microcode load.

inband
: An option allowing FlashCopy requests to be issued remotely through an existing PPRC link.

incremental
: An option providing FlashCopy the capability to refresh a volume when only a subset of data has changed, reducing background copy time.

initial microcode load (IML)
: The act of loading microcode.

I/O device
: An addressable input/output unit, such as a direct access storage device, magnetic tape device, or printer.

IPL Initial program load.

JCL Job control language.

Job control language (JCL)
: A problem-oriented language used to identify the job or describe its requirements to an operating system.

journal data set
: A checkpoint data set that contains work to be done. For XRC, the work to be done consists of all changed records from the primary volumes. Changed records are collected and formed into a “consistency group”, and then the group of updates is applied to the secondary volumes.

KB Kilobyte.

keyword
: A name that identifies a parameter in a command string. Keywords can be entered in their entirety or as abbreviations identified in the syntax diagram for the command.

kilobyte (KB)
: 1 024 bytes.

kilometer (km)
: One thousand meters; 0.62 mile.

km Kilometer.

LIC Licensed Internal Code.

logical storage subsystem
: A collection of addresses that are associated with the same logical subsystem.

LOGPLUS
: A facility providing mirroring of data written by the z/OS Logger.

LPAR *L*ogical *PAR*tition. A logical segmentation of a mainframe's memory and other resources that allows it to run its own copy of the operating system and associated applications. See also *virtual server.*

LSS Logical storage subsystem.

master journal time
: The minimum of the journal times reported by all logical sessions. This is the latest time to which all sessions can recover their data in the event of a disaster, and still provide coupled consistency. Because coupled consistency requires all volumes to be at the same time, no session is allowed to apply data that is after the master journal time.

master data set
: The master data set ensures consistency among all XRC subsystems contained within the coupled XRC system

master session
: A logical entity that is used to coordinate session commands and data consistency across multiple XRC sessions. A master session exists as long as there is an XRC session coupled to the master session.

MB Megabyte.

megabyte (MB)
1 048 576 bytes.

metro mirroring
Peer-to-Peer Remote Copy, or PPRC.

MIP Million instructions per second.

multiple data mover
A function of Coupled Extended Remote Copy (CXRC) that allows several logical system data mover sessions that are running on independent MVS system images to be logically connected so that all volumes in all sessions are treated as if they are part of the same system data mover session. This allows a system data mover configuration to grow to support more volumes than can be supported by a single MVS image.

multiple extended remote copy (MXRC)
An enhancement to XRC that allows you to run up to five XRC sessions within a single LPAR.

Multiple Virtual Storage (MVS)
One of a family of IBM operating systems for the System/370 or System/390 processor, such as z/OS.

MVS Multiple Virtual Storage.

nonvolatile storage (NVS)
Random access electronic storage with a backup battery power source, used to retain data during a power failure. Nonvolatile storage, accessible from all cached IBM storage clusters, stores data during DASD fast write, dual copy, and remote copy operations.

NVS Nonvolatile storage.

operating system
Software that controls the execution of programs. An operating system may provide services such as resource allocation, scheduling, input/output control, and data management.

orphan data
Data that occurs between the last, safe backup for a recovery system and the time when the application system experiences a disaster. This data is lost when either the application system becomes available for use or when the recovery system is used in place of the application system.

parallel access volume (PAV)
Parallel access volume.

partitioned data set extended (PDSE)
A system-managed, page-formatted data set on direct access storage.

P/DAS
PPRC dynamic address switching.

PDSE Partitioned data set extended.

peer-to-peer remote copy (PPRC)
A hardware-based remote copy option that provides a synchronous volume copy across storage subsystems for disaster recovery, device migration, and workload migration.

peer-to-peer remote copy (PPRC)-extended distance(XD)
In this PPRC mode, the volume pair does not reach DUPLEX state after the initial copy of the volume pair is completed. The volume remains in a PENDING.XD state. The primary volume updates complete before they are mirrored to the secondary volume.

pending
The initial state of a defined volume pair, before it becomes a duplex pair. During this state, the contents of the primary volume are copied to the secondary volume.

pinned data
Data that is held in a cached storage control, because of a permanent error condition, until it can be destaged to DASD or until it is explicitly discarded by a host command. Pinned data exists only when using fast write, dual copy, or remote copy functions.

port (1) An access point for data entry or exit. (2) A receptacle on a device to which a cable for another device is attached.

PPRC Peer-to-peer remote copy.

PPRC Dynamic Address Switching (P/DAS)
A software function that provides the ability to dynamically redirect all application I/O from one PPRC volume to another PPRC volume.

PPRC-XD
Peer-to-peer remote copy-extended distance.

primary device
One device of a dual copy or remote copy volume pair. All channel commands to the copy logical volume are directed to the primary device. The data on the primary device is duplicated on the secondary device. See also *secondary device*.

primary system
A system made up of one or more host systems that perform the main set of functions for an establishment. This is the system that updates the primary Disk volumes that are being copied by a copy services function. Also referred to as application system.

PTF Program temporary fix.

quiesce
To render a device temporarily inactive or disabled.

RACF Resource access control facility.

RAS Retrievability, availability, and serviceability.

read hit
When data requested by the read operation is in the cache.

read miss
When data requested by the read operation is not in the cache.

recover
To rebuild data after it has been damaged or destroyed.

recovery
The process of rebuilding data after it has been damaged or destroyed. In the case of remote copy, this involves applying data from secondary volume copies.

recovery system
A system that is used in place of a primary application system that is no longer available for use. Data from the application system must be available for use on the recovery system. This is usually accomplished through backup and recovery techniques, or through various DASD copying techniques, such as remote copy.

remote copy
A storage-based disaster recovery and workload migration function that can copy data in real time to a remote location. Two options of remote copy are available. See *peer-to-peer remote copy* and *extended remote copy*.

restore
Synonym for recover.

resynchronization
A track image copy from the primary volume to the secondary volume of only the tracks which have changed since the volume was last in duplex mode.

RVA RAMAC Virtual Array Storage Subsystem.

SAID System adapter identification.

SAM Sequential access method.

SDM System data mover.

secondary device
One of the devices in a dual copy or remote copy logical volume pair that contains a duplicate of the data on the primary device. Unlike the primary device, the secondary device may only accept a limited subset of channel commands.

service information message (SIM)
A message, generated by a storage subsystem, that is the result of error event collection and analysis. A SIM indicates that some service action is required.

sidefile
A storage area used to maintain copies of tracks within a concurrent copy domain. A concurrent copy operation maintains a sidefile in storage control cache and another in processor storage.

SIM Service information message.

simplex state
A volume is in the simplex state if it is not part of a dual copy or a remote copy volume pair. Ending a volume pair returns the two devices to the simplex state. In this case, there is no longer any

capability for either automatic updates of the secondary device or for logging changes, as would be the case in a suspended state.

SMF System Management Facilities.

SMS Storage Management Subsystem, part of DFSMSdfp.

SRM System resources manager.

snapshot copy
A point-in-time copy services function that can quickly copy data from a source location to a target location.

SSID Subsystem identifier.

stage The process of writing data from a DASD to the cache.

state data set
A data set that contains status of the XRC session and of associated volumes that XRC is managing. The state data set is updated XADDPAIR, XDELPAIR, XSET, XSUSPEND, XRECOVER, or XEND command is issued, or whenever a volume state changes.

storage cluster
A power and service region that runs channel commands and controls the storage devices. Each storage cluster contains both channel and device interfaces. Storage clusters also perform the DASD control functions.

storage control
The component in a storage subsystem that handles interaction between processor channel and storage devices, runs channel commands, and controls storage devices.

STORAGE_CONTROL_DEFAULT
A specification used by several XRC commands and messages to refer to the timeout value specified in the maintenance panel of the associated storage control.

storage control session
A logical entity that is created for the purpose of processing updates to the XRC primary volumes. It is used to group sets of primary XRC volumes that are being processed by an XRC session within the storage control.

storage director
In an IBM storage control, a logical entity consisting of one or more physical storage paths in the same storage cluster. See also *storage path*.

Storage Management Subsystem (SMS)
A component of MVS/DFP that is used to automate and centralize the management of storage by providing the storage administrator with control over data class, storage class, management class, storage group, aggregate group and automatic class selection routine definitions.

storage path
The hardware within the IBM storage control that transfers data between the DASD and a channel. See also *storage director*.

storage subsystem
A storage control and its attached storage devices.

string A series of connected DASD units sharing the same A-unit (or head of string).

subsystem
See *DASD subsystem* or *storage subsystem*.

subsystem identifier (SSID)
A user-assigned number that identifies a DASD subsystem. This number is set by the service representative at the time of installation and is included in the vital product data.

super consistency group
A set of consistency groups that were combined during secondary processing.

suspended state
When only one of the devices in a dual copy or remote copy volume pair is being updated because of either a permanent error condition or an authorized user command. All writes to the remaining functional device are logged. This allows for automatic resynchronization of both volumes when the volume pair is reset to the active duplex state.

synchronization
An initial volume copy. This is a track image copy of each primary track on the volume to the secondary volume.

synchronous operation
A type of operation in which the remote copy PPRC function copies updates to the

secondary volume of a PPRC pair at the same time that the primary volume is updated. Contrast with *asynchronous operation*.

sysplex
A set of MVS or z/OS systems that are communicating and cooperating with each other through certain multisystem hardware components and software services, such as CXRC, to process workloads. This term is derived from "system complex".

system data mover
A system that interacts with storage controls that have attached XRC primary volumes. The system data mover copies updates made to the XRC primary volumes to a set of XRC-managed secondary volumes.

system-managed data set
A data set that has been assigned a storage class.

TOD Time of day.

Time Sharing Option (TSO)
A System/370 operating system option that provides interactive time sharing from remote terminals.

timeout
The time in seconds that the storage control remains in a "long busy" condition before physical sessions are ended.

timestamp
The affixed value of the system time-of-day clock at a common point of reference for all write I/O operations directed to active XRC primary volumes. The UTC format in this book is *yyyy.ddd hh:mm:ss.thmiju*.

TSO Time Sharing Option.

Universal Time, Coordinated
Used as a global time reference. The format in this book is *yyyy.ddd hh:mm:ss.thmiju*.

utility volume
A volume that is available to be used by the extended remote copy function to perform system data mover I/O for a primary site storage control's XRC-related data.

UTC Universal Time, Coordinated.

vital product data (VPD)
Nonvolatile data that is stored in various locations in the DASD subsystem. It includes configuration data, machine serial number, and machine features.

virtual server
A logical partition.

volser Volume serial number.

volume
The DASD space identified by a common serial number and accessed by any of a set of related addresses. See also *device*.

volume level FlashCopy
FlashCopy of one complete source volume to a target volume.

VPD Vital product data.

VSAM
Virtual storage access method.

VTOC Volume table of contents.

workload migration
The process of moving an application's data from one set of DASD to another for the purpose of balancing performance needs, moving to new hardware, or temporarily relocating data.

write hit
A write operation where the data requested is in the cache.

write miss
A write operation where the data requested is not in the cache.

write update
A write operation that updates a direct access volume.

XDF Extended distance feature (of ESCON).

XRC Extended remote copy.

XRC planned-outage-capable
A storage subsystem with an LIC level that supports a software bitmap but not a hardware bitmap.

XRC session
A logical entity that is created for the purpose of processing XRC commands and coordinating the movement of data between primary and secondary volumes.

XRC unplanned-outage-capable
A storage subsystem with an LIC level that supports a hardware bitmap.

Index

Numerics

A

B

C

D

E

F

K

Q

R

T

U

V

W

Product Number: 5650-ZOS

Printed in USA

SC23-6847-02