Designing Disaster Tolerant High Availability Clusters
Legal Notices

The information contained in this document is subject to change without notice.

Hewlett-Packard makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.


This document contains information which is protected by copyright. All rights are reserved. Reproduction, adaptation, or translation without prior written permission is prohibited, except as allowed under the copyright laws.

Corporate Offices:

Hewlett-Packard Co.
3000 Hanover St.
Palo Alto, CA 94304

Use, duplication or disclosure by the U.S. Government Department of Defense is subject to restrictions as set forth in paragraph (b)(3)(ii) of the Rights in Technical Data and Software clause in FAR 52.227-7013.

Rights for non-DOD U.S. Government Departments and Agencies are as set forth in FAR 52.227-19(c)(1,2).

Use of this manual and flexible disc(s), compact disc(s), or tape cartridge(s) supplied for this pack is restricted to this product only. Additional copies of the programs may be made for security and back-up purposes only. Resale of the programs in their present form or with alterations, is expressly prohibited.

A copy of the specific warranty terms applicable to your Hewlett-Packard product and replacement parts can be obtained from your local Sales and Service Office.

EMC², Symmetrix, PowerPath, and SRDF are trademarks of EMC Corporation.
Contents

1. Disaster Tolerance and Recovery in an MC/ServiceGuard Cluster
   Evaluating the Need for Disaster Tolerance ......................... 14
   What is a Disaster Tolerant Architecture?......................... 16
   Types of Disaster Tolerant Clusters ................................. 18
      Local Cluster .................................................... 18
      Campus Clusters ................................................ 18
      Metropolitan Cluster .......................................... 20
      Continental Cluster ............................................. 22
      Continental Cluster With Cascading Failover ................. 23
   Disaster Tolerant Architecture Guidelines ...................... 25
      Protecting Nodes through Geographic Dispersion ............. 25
      Protecting Data through Replication .......................... 26
      Using Alternative Power Sources ............................... 32
      Creating Highly Available Networking ....................... 33
   Disaster Tolerant Cluster Limitations ......................... 37
   Managing a Disaster Tolerant Environment .................... 39
   Using this Guide with Your Disaster Tolerant Cluster Products 41

2. Building a Campus Cluster Using FibreChannel and MC/ServiceGuard
   Guidelines for Disaster Tolerant Architectures with FibreChannel .... 44
   Two Data Centers with FibreChannel Hubs ....................... 46
   Two Data Centers with FibreChannel Point-to-Point ........... 48
   Advantages and Disadvantages of a Two-Data-Center Architecture .. 49
3. Building a Metropolitan Cluster Using MetroCluster/CA

Overview of MetroCluster/CA .................................................. 52
Designing a Disaster Tolerant Architecture for use with
MetroCluster/CA ............................................................ 53
Single Data Center ............................................................ 53
Three Data Centers with Arbitrator(s) .................................. 53
Disaster Tolerant Checklist .................................................. 61
Cluster Configuration Worksheet ......................................... 62
Package Configuration Worksheet ....................................... 63
Preparing an MC/ServiceGuard Cluster for MetroCluster/CA .... 64
Setting up the Hardware ..................................................... 64
Setting Fence Levels ......................................................... 65
Installing the Necessary Software ......................................... 69
Creating the Raid Manager Configuration ............................. 70
Sample Raid Manager Configuration File ............................... 73
Configuring Automatic Raid Manager Startup ....................... 75
Verifying the XP Series Disk Array Configuration ................... 76
Creating and Exporting Volume Groups ............................... 77
Importing Volume Groups on Other Nodes ........................... 78
Configuring PV Links .......................................................... 78
Configuring Packages for Automatic Disaster Recovery ............ 79
Maintaining a Cluster that uses MetroCluster/CA ................... 83
Viewing the Progress of Copy Operations ............................. 83
Viewing Side File Size ....................................................... 83
Normal Maintenance ......................................................... 84
Resynchronizing ............................................................... 84
Using the pairresync Command ......................................... 86
Failback ........................................................................ 86
4. Building a Metropolitan Cluster Using MetroCluster/SRDF

Overview of MetroCluster/SRDF .................................................. 88

Designing a Disaster Tolerant Architecture for use with
MetroCluster/SRDF ........................................................................ 90
Single Data Center ...................................................................... 90
Three Data Centers with Arbitrator(s) ......................................... 91
Disaster Tolerant Checklist ......................................................... 97
Cluster Configuration Worksheet ................................................. 98
Package Configuration Worksheet ............................................... 99

Preparing a Cluster for MetroCluster/SRDF .............................. 100
Installing the Necessary Software .............................................. 100
Setting up Hardware for 1 by 1 Configurations ............................ 101
Setting up Hardware for M by N Configurations .......................... 102
Grouping the Symmetrix Devices at Each Data Center ............... 104
Building the Symmetrix CLI Database ....................................... 105
Determining Symmetrix Device Names on Each Node ............... 106

Setting up 1 by 1 Configurations .............................................. 112
Creating Symmetrix Device Groups .......................................... 112
Configuring Gatekeeper Devices .............................................. 114
Verifying the EMC Symmetrix Configuration ........................... 115
Creating and Exporting Volume Groups ................................... 115
Importing Volume Groups on Other Nodes ............................... 115
Configuring PV Links ............................................................... 116

Setting up M by N Configurations .............................................. 117
Creating Symmetrix Device Groups .......................................... 118
Configuring Gatekeeper Devices .............................................. 119
Defining the Consistency Groups .............................................. 120
Creating Volume Groups .......................................................... 121

Configuring MC/ServiceGuard Packages for Automatic
Disaster Recovery ........................................................................ 123
# Contents

Maintaining a Cluster that Uses MetroCluster/SRDF .......................... 128
Managing Business Continuity Volumes ........................................ 129
Protecting against Rolling Disasters .......................................... 129
Using the BCV in Resynchronization .......................................... 129

5. Building a Continental Cluster

Understanding Continental Cluster Concepts .............................. 132
Mutual Recovery Configuration .................................................. 133
Application Recovery in a Continental Cluster ............................ 134
Monitoring over a Wide Area Network ...................................... 135
Cluster Events ........................................................................... 136
Interpreting the Significance of Cluster Events ............................ 138
How Notifications Work ............................................................. 138
Alerts ...................................................................................... 139
Alarms ..................................................................................... 139
Creating Notifications for Failure Events ................................. 139
Creating Notifications for Events that Indicate a Return of Service 140
Performing Cluster Recovery ...................................................... 140
Notes on Packages in a Continental Cluster ......................... 141
How MC/ServiceGuard commands work in a ContinentalCluster 142

Designing a Disaster Tolerant Architecture for use with
ContinentalClusters ................................................................. 145
Mutual Recovery ................................................................. 145
MC/ServiceGuard Clusters ...................................................... 145
Data Replication ................................................................. 146
Highly Available Wide Area Networking .................................. 148
Data Center Processes ............................................................ 148
ContinentalClusters Worksheets .............................................. 149

Preparing the Clusters ............................................................... 153
Setting up and Testing Data Replication ................................. 153
Contents

Configuring a Cluster without Recovery Packages .................... 154
Configuring a Cluster with Recovery Packages ....................... 156

Building the ContinentalClusters Configuration ..................... 159
Preparing Security Files ............................................. 159
Creating the Monitor Package ....................................... 162
Editing the ContinentalClusters Configuration File .................. 163
Checking and Applying the ContinentalClusters Configuration ..... 176
Starting the ContinentalClusters Monitor Package ................. 178
Validating the Configuration ........................................ 179
Documenting the Recovery Procedure ................................ 180
Reviewing the Recovery Procedure ................................... 181

Testing the Continental Cluster ....................................... 182
Testing Individual Packages .......................................... 182
Testing ContinentalClusters Operations .............................. 183

Switching to the Recovery Packages in Case of Disaster .......... 185
Receiving Notification ................................................ 185
Verifying that Recovery is Needed .................................. 186
Using the Recovery Command to Switch All Packages ............. 186
How the cmrecovercl Command Works ............................... 188

Restoring Disaster Tolerance ......................................... 190
Restore Clusters to their Original Roles ............................ 190
Primary Packages Remain on the Surviving Cluster ............... 191
Newly Created Cluster Will Run Primary Packages ............... 192
Newly Created Cluster Will Function as Recovery Cluster for All Recovery Groups .......................... 193

Maintaining a Continental Cluster ................................... 194
Adding a Node to a Cluster or Removing a Node from a Cluster ... 194
Adding a Package to the Continental Cluster ....................... 195
Removing a Package from the Continental Cluster ................. 196
Changing Monitoring Definitions .................................... 196
Contents

Checking the Status of Clusters, Nodes, and Packages .......... 197
Reviewing Messages and Log Files .......................... 200
Deleting a Continental Cluster Configuration ................. 201
Renaming a Continental Cluster ............................... 202
Checking Java Versions ....................................... 202

6. Physical Data Replication for Continental Clusters
Using Continuous Access XP

Overview ....................................................... 204
Preparing the Continental Cluster for Data Replication ......... 206
Notes on the Raid Manager Configuration ...................... 210
Creating and Exporting Volume Groups ......................... 211
Setting up a Primary Package on the Primary Cluster ........... 213
Setting up a Recovery Package on the Recovery Cluster ....... 217
Setting up the Continental Cluster Configuration ............... 221
Switching to the Recovery Cluster in Case of Disaster ......... 223
Failback Scenarios ........................................... 224
Scenario 1 ...................................................... 224
Scenario 2 ...................................................... 224
Failback in Scenarios 1 and 2 ................................ 224
Scenario 3 ...................................................... 226
Failback in Scenario 3 ......................................... 226
Failback When the Primary Has SMPL Status ................... 228

Maintaining the Continuous Access XP Data Replication
Environment ..................................................... 229
Rescynchronizing ............................................... 229
Using the pairresync Command ................................ 231
Some Further Points ........................................... 231
Contents

7. Physical Data Replication for Continental Clusters Using EMC SRDF
   Overview .......................................................... 234
   Preparing a Continental Cluster for Data Replication ............ 235
   Creating and Exporting Volume Groups ....................... 241
   Setting up a Primary Package on the Primary Cluster .......... 244
   Setting up a Recovery Package on the Recovery Cluster ....... 248
   Setting up the Continental Cluster Configuration ............. 251
   Switching to the Recovery Cluster in Case of Disaster ....... 253
   Failback Scenarios ............................................... 254
      Scenario 1 ..................................................... 254
      Scenario 2 ..................................................... 258
   Maintaining the EMC SRDF Data Replication Environment .... 260
      Normal Startup ............................................... 260
      Normal Maintenance ......................................... 261
      Some Further Points ......................................... 261

8. Cascading Failover in a Continental Cluster
   Overview .......................................................... 266
   Symmetrix Configuration ....................................... 267
   Using Template Files .......................................... 268
   Data Storage Setup ............................................. 269
      Setting Up Symmetrix Device Groups ...................... 269
      Setting up Volume Groups .................................. 274
      Testing the Volume Groups .................................. 276
   Primary Cluster Package Setup ................................ 278
   Recovery Cluster Package Setup ............................... 279
Contents

Continental Cluster Configuration ................................. 280
Data Replication Procedures ..................................... 281
  Data Initialization Procedures ................................. 281
  Data Refresh Procedures in the Steady State ............... 283
  Data Replication in Failover and Failback Scenarios .... 285

A. Package Control Script Variables for MetroCluster/CA

B. Package Control Script Variables for MetroCluster/SRDF

C. Configuration File Parameters for ContinentalClusters

D. ContinentalClusters Command and Daemon Reference

  Glossary
Printing History

Table 1

<table>
<thead>
<tr>
<th>Printing Date</th>
<th>Part Number</th>
<th>Edition</th>
<th>Operating System Releases (see Note below)</th>
</tr>
</thead>
<tbody>
<tr>
<td>June 1998</td>
<td>B6264-90002</td>
<td>Edition 1</td>
<td>HP-UX 10.20 and 11.0</td>
</tr>
<tr>
<td>August 1999</td>
<td>B7660-90001</td>
<td>Edition 2</td>
<td>HP-UX 10.20 and 11.0</td>
</tr>
<tr>
<td>December 1999</td>
<td>B7660-90002</td>
<td>Edition 3</td>
<td>HP-UX 10.20 and 11.0</td>
</tr>
<tr>
<td>February 2000</td>
<td>B7660-90003</td>
<td>Edition 4</td>
<td>HP-UX 10.20 and 11.00</td>
</tr>
<tr>
<td>September 2000</td>
<td>B7660-90005</td>
<td>Edition 5</td>
<td>HP-UX 10.20, 11.0 and 11i</td>
</tr>
<tr>
<td>March 2001</td>
<td>B7660-90006</td>
<td>Edition 6</td>
<td>HP-UX 10.20, 11.0 and 11i</td>
</tr>
</tbody>
</table>

The printing date and part number indicate the current edition. The printing date changes when a new edition is printed. (Minor corrections and updates which are incorporated at reprint do not cause the date to change.) The part number changes when extensive technical changes are incorporated. New editions of this manual will incorporate all material updated since the previous edition.

NOTE

This document describes a group of separate software products that are released independently of one another. Not all products described in this document are necessarily supported on all the same operating system releases. Consult your product’s Release Notes for information about supported platforms.

HP Printing Division:
Business Critical Computing Business Unit
Hewlett-Packard Co.
19111 Pruneridge Ave
Cupertino, CA 95014
Chapter 1

Disaster Tolerance and Recovery in an MC/ServiceGuard Cluster

This guide shows how to use a variety of Hewlett-Packard high availability cluster technologies to provide disaster tolerance for your mission-critical applications. It is assumed that you are already familiar with MC/ServiceGuard high availability concepts and configurations.

This chapter introduces the basic concepts of disaster tolerance and disaster recovery within the context of an MC/ServiceGuard cluster. It covers:

• Evaluating the Need for Disaster Tolerance
• What is a Disaster Tolerant Architecture?
• Types of Disaster Tolerant Clusters
• Disaster Tolerant Architecture Guidelines
• Managing a Disaster Tolerant Environment
• Using this Guide with Your Disaster Tolerant Cluster Products
Disaster Tolerance and Recovery in an MC/ServiceGuard Cluster

Evaluating the Need for Disaster Tolerance

Disaster tolerance is the ability to restore applications and data within a reasonable period of time after a disaster. Most think of fire, flood, and earthquake as disasters, but a disaster can be any event that unexpectedly interrupts service or corrupts data in an entire data center: the backhoe that digs too deep and severs a network connection, or an act of sabotage.

Disaster tolerant architectures protect against unplanned down time due to disasters by geographically distributing the nodes in a cluster so that a disaster at one site does not disable the entire cluster. To evaluate your need for a disaster tolerant solution, you need to weigh:

- Risk of disaster. Areas prone to tornadoes, floods, or earthquakes may require a disaster recovery solution. Some industries need to consider risks other than natural disasters or accidents, such as terrorist activity or sabotage.

  The type of disaster to which your business is prone, whether it is due to geographical location or the nature of the business, will determine the type of disaster recovery you choose. For example, if you live in a region prone to big earthquakes, you are not likely to put your alternate or backup nodes in the same city as your primary nodes, because that sort of disaster affects a large area.

  The frequency of the disaster also plays an important role in determining whether to invest in a rapid disaster recovery solution. For example, you would be more likely to protect from hurricanes that happen every season, rather than protecting from a volcano that hasn't erupted in 100 years.

- Vulnerability of the business. How long can your business afford to be down? Some parts of a business may be able to endure a 1 or 2 day recovery time, while others need to recover in a matter of minutes. Some parts of a business only need local protection from single outages, such as a node failure. Other parts of a business may need both local protection and protection in case of site failure.

  It is important to consider the role the data servers play in your business. For example, you may target the assembly line production servers as most in need of quick recovery. But if the most likely disaster in your area is an earthquake, it would render the assembly
Disaster Tolerance and Recovery in an MC/ServiceGuard Cluster

Evaluating the Need for Disaster Tolerance

line inoperable as well as the computers. In this case disaster recovery would be moot, and local failover is probably the more appropriate level of protection.

On the other hand, you may have an order processing center that is prone to floods in the winter. The business loses thousands of dollars a minute while the order processing servers are down. A disaster tolerant architecture is appropriate protection in this situation.

Deciding to implement a disaster recovery solution really depends on the balance between risk of disaster, and the vulnerability of your business if a disaster occurs. The following pages give a high-level view of a variety of disaster tolerant solutions and sketch the general guidelines that you should follow in developing a disaster tolerant computing environment.
What is a Disaster Tolerant Architecture?

In an MC/ServiceGuard cluster configuration, high availability is achieved by using redundant hardware to eliminate single points of failure. This protects the cluster against hardware faults, such as the node failure in Figure 1-1.

Figure 1-1  High Availability Architecture.

For some installations, this level of protection is insufficient. Consider the order processing center where power outages are common during harsh weather. Or consider the systems running the stock market, where multiple system failures, for any reason, have a significant financial
Disaster Tolerance and Recovery in an MC/ServiceGuard Cluster

What is a Disaster Tolerant Architecture?

impact. For these types of installations, and many more like them, it is important to guard not only against single points of failure, but against multiple points of failure (MPOF), or against single massive failures that cause many components to fail, such as the failure of a data center, of an entire site, or of a small area. A data center, in the context of disaster recovery, is a physically proximate collection of nodes and disks, usually all in one room.

Creating clusters that are resistant to multiple points of failure or single massive failures requires a different type of cluster architecture called a disaster tolerant architecture. This architecture provides you with the ability to fail over automatically to another part of the cluster or manually to a different cluster after certain disasters. Specifically, the disaster tolerant cluster provides appropriate failover in the case where a disaster causes an entire data center to fail.

Figure 1-2 Disaster Tolerant Architecture

Data Center A

node 1

pkg A

node 2

pkg B

Data Center B

node 3

pkg C

node 4

pkg D

Client Connections

Packages A and B fail over to Data Center B

Data Center A

node 1

node 2

Data Center A fails

Client Connections

Data Center B
Types of Disaster Tolerant Clusters

To protect against multiple points of failure, a cluster must be geographically dispersed: nodes can be put in different rooms, on different floors of a building, or even in separate buildings or separate cities. The distance between the nodes is dependent on the types of disaster from which you need protection, and on the technology used to replicate data.

Local Cluster

A local cluster has all nodes located in a single data center and is not disaster tolerant. Because most high-availability clusters are local clusters, this type is included in the discussion as a baseline for comparison with other cluster architectures.

Campus Clusters

A campus cluster has alternate nodes located in different data centers. These data centers can be separate rooms in a building, adjacent buildings, or even buildings separated by some distance. The word “campus” implies that the organization housing the cluster owns or
Disaster Tolerance and Recovery in an MC/ServiceGuard Cluster

Types of Disaster Tolerant Clusters

leases the land and buildings such that no outside permission is needed to dig trenches, lay cable, or reroute power circuits. Campus clusters are designed so that no single building failure will cause the cluster to fail.

Campus clusters are connected using a high speed cable that guarantees network access between the nodes as long as all guidelines for disaster tolerant architecture are followed. The distance between nodes in a campus cluster is limited by the data replication technology.

Figure 1-4 Campus Cluster

Architecture requirements for campus clusters usually consist of:

• Redundant disks using data replication. An example of this is HP’s MirrorDisk/UX, which replicates data to two disks of any type.

NOTE

While it is possible to configure physical data replication through products such as HP’s XP Series disk arrays with Continuous Access XP or Symmetrix EMC SRDF, it is still necessary to provide for high availability at the local level through RAID or mirroring.

• Redundant network cables installed using different routes. This protects the cluster from a single accident severing both network cables at once.

• Power for each data center supplied from different power circuits. Some may even want to lease redundant power from different substations. This protects the cluster from a single power failure due
Disaster Tolerance and Recovery in an MC/ServiceGuard Cluster

Types of Disaster Tolerant Clusters

• to shorts or accidents that cuts power to all nodes in the cluster.

• A campus cluster with four nodes must use dual cluster locks to provide a tie-breaker in case heartbeat is lost among the nodes. As an alternative, the campus cluster can use arbitrators if it follows the same rules as the metropolitan cluster designs (described in Chapters 3 and 4). However, the conventional four-node campus cluster shown in Chapter 2 uses dual lock disks.

Campus clusters are implemented use standard MC/ServiceGuard with FibreChannel mass storage.

Metropolitan Cluster

A metropolitan cluster is a cluster that has alternate nodes located in different parts of a city or in adjacent cities. Putting nodes further apart increases the likelihood that alternate nodes will be available for failover in the event of a disaster. Metropolitan clusters are very similar to campus clusters with the exception that metropolitan clusters often require right-of-way from local governments or utilities to lay network and data replication cable. This can complicate the design and implementation.

Metropolitan clusters also require a different kind of tie-breaker system for ensuring that split-brain situations do not arise. Typically, metropolitan clusters use an arbitrator site containing additional cluster nodes instead of the cluster lock disk.

The architectural requirements are the same as for a campus cluster, with the additional constraint of the third data center. And as with a campus cluster, the distance separating the nodes in a metropolitan cluster is limited by the data replication and network technology available. Metropolitan cluster architecture is implemented through two HP products:

• MetroCluster with Continuous Access XP (described in Chapter 3)
• MetroCluster with EMC SRDF (described in Chapter 4)
A key difference between campus and metropolitan clusters is the data replication technology used. The campus cluster uses FibreChannel and MirrorDisk/UX, which limits the distances between data centers. Metropolitan clusters use data replication based on the capabilities of the HP SureStore E Disk Array XP series or the EMC Symmetrix array, which allow greater distances.
Disaster Tolerance and Recovery in an MC/ServiceGuard Cluster

Types of Disaster Tolerant Clusters

**Continental Cluster**

A **continental cluster** provides alternate clusters that are separated by large distances so that wide area networking is used between them. Continental cluster architecture is implemented via the ContinentalClusters product, described fully in Chapter 5. The design is implemented with two distinct ServiceGuard clusters located in different geographic areas. In this architecture, each cluster maintains its own quorum, so an arbitrator data center is not used. ContinentalClusters is architected to use any WAN connection via a TCP/IP protocol; however, due to data replication needs, high speed connections such as T1 or T3/E3 leased lines or switched lines may be required.

![Continental Cluster Diagram](image)

The key issues concerning a WAN cluster are:

- Inter-cluster connections for ContinentalClusters are TCP/IP based connections.
- The physical connection is one or more leased lines managed by a common carrier. Common carriers cannot guarantee the same reliability that a dedicated physical cable can. The distance can introduce a time lag for data replication, which creates an issue with data currency. This could increase the cost by requiring higher speed...
Disaster Tolerance and Recovery in an MC/ServiceGuard Cluster

Types of Disaster Tolerant Clusters

WAN connections to improve data replication performance and reduce latency.

- Tools such as Transaction Processing Monitors or database replication tools that work across a WAN are needed to make sure the data replication maintains data consistency.
- Operational issues, such as working with different staff with different processes, and conducting failover rehearsals, are made more difficult the further apart the nodes in the cluster are.

Continental Cluster With Cascading Failover

Another way of setting up your continental cluster is the use of cascading failover, an option available for users of the EMC Symmetrix. Cascading failover means that applications are configured to fail over on nodes within two data centers in either cluster. They fail over to a cluster if the entire alternate cluster is down. For example, bi-directional failover allows a cluster to be both a recovery cluster and a primary cluster for different packages. Data replication also follows the cascading model. Data is replicated from the primary disk array to the secondary disk array in the MetroCluster, then data is replicated to the third disk array in the ServiceGuard recovery cluster.

A continental cluster with cascading failover uses four data centers distributed between a metropolitan cluster, which serves as a primary cluster, and a standard cluster, which serves as a recovery cluster. The configuration uses three EMC Symmetrix frames, two of which are part of the metropolitan cluster and the other attached to the recovery cluster.

The data centers are distributed as follows:

- Primary Symmetrix—on the site that holds the primary copy of the data, located in the primary cluster.
- Secondary Symmetrix—on the site that holds a remote mirror copy of the data, located in the primary cluster.
- Arbitrator—a separate site that contains the arbitrator nodes, located in the primary cluster.
- Recovery Symmetrix—on a site that holds a remote mirror copy of the data, located in the recovery cluster.
Disaster Tolerance and Recovery in an MC/ServiceGuard Cluster

Types of Disaster Tolerant Clusters

Figure 1-7 illustrates data centers, clusters, nodes and Symmetrix frames in a cascading failover configuration. Refer to Chapter 8 for details on setting up data replication for this type of cluster.

Figure 1-7 Cascading Failover Data Center Distribution
Disaster Tolerant Architecture Guidelines

Disaster tolerant architectures represent a shift away from the massive central data centers towards more distributed data processing facilities. While each architecture will be different to suit specific availability needs, there are a few basic guidelines for designing a disaster tolerant architecture so that it protects against the loss of an entire data center:

- Protecting nodes through geographic dispersion
- Protecting data through replication
- Using alternative power sources
- Creating highly available networks

These guidelines are in addition to the standard high-availability guidelines of redundant components such as PV links, network cards, power supplies, and disks.

Protecting Nodes through Geographic Dispersion

Redundant nodes in a disaster tolerant architecture must be geographically dispersed. If they are in the same data center, it is not a disaster tolerant architecture. Figure 1-2 on page 17 shows a cluster architecture with nodes in two data centers: A and B. If all nodes in data center A fail, applications can fail over to the nodes in data center B and continue to provide clients with service.

Depending on the type of disaster you are protecting against and on the available technology, the nodes can be as close as another room in the same building, or as far away as another city. The minimum recommended dispersion is a single building with redundant nodes in different data centers using different power sources. Specific architectures based on geographic dispersion are discussed in “Types of Disaster Tolerant Clusters” on page 18.
Disaster Tolerance and Recovery in an MC/ServiceGuard Cluster

Disaster Tolerant Architecture Guidelines

Protecting Data through Replication

The most significant losses during a disaster are the loss of access to data, and the loss of data itself. You protect against this loss through data replication, that is, creating extra copies of the data. Data replication should:

- Ensure **data consistency** by replicating data in a logical order so that it is immediately usable or recoverable. Inconsistent data is unusable and is not recoverable for processing. Consistent data may or may not be current.

- Ensure **data currency** by replicating data quickly so that a replica of the data can be recovered to include all committed transactions that were applied to the local database.

- Ensure **data recoverability** so that there is some action that can be taken to make the data consistent, such as restoring from backup.

- Minimize **data loss** by configuring data replication to address consistency, currency, and recoverability.

Different data replication methods have different advantages with regards to data consistency and currency. Your choice of which data replication methods to use will depend on what type of disaster tolerant architecture you require.

Off-line Data Replication

Off-line data replication is the method most commonly used today. It involves two or more data centers that store their data on tape and either send it to each other (via an express service, if need dictates) or store it off-line in a vault. If a disaster occurs at one site, the off-line copy of data is used to synchronize data and a remote site functions in place of the failed site.

Because data is replicated using physical off-line backup, data consistency is fairly high, barring human error or an untested corrupt backup. However, data currency is compromised by the time delay in sending the tape backup to a remote site.

Off-line data replication is fine for many applications for which recovery time is not an issue critical to the business. Although data might be replicated weekly or even daily, recovery could take from a day to a week depending on the volume of data. Some applications, depending on the role they play in the business, may need to have a faster recovery time,
within hours or even minutes.

**On-line Data Replication**

On-line data replication is a method of copying data from one site to another across a link. It is used when very short recovery time, from minutes to hours, is required. To be able to recover use of a system in a short time, the data at the alternate site must be replicated in real time on all disks.

Data can be replicated either synchronously or asynchronously. **Synchronous replication** requires one disk write to be completed and replicated before another disk write can begin. This method improves the chances of keeping data consistent and current during replication. However, it greatly reduces replication capacity and performance, as well as system response time. **Asynchronous replication** does not require the primary site to wait for one disk write to be replicated before beginning another. This can be an issue with data currency, depending on the volume of transactions. An application that has a very large volume of transactions can get hours or days behind in replication using asynchronous replication. If the application fails over to the remote site, it would start up with data that is not current.

Currently the two ways of replicating data on-line are physical data replication and logical data replication. Either of these can be configured to use synchronous or asynchronous writes.

**Physical Data Replication** Each physical write to disk is replicated on another disk at another site. Because the replication is a physical write to disk, it is not application dependent. This allows each node to run different applications under normal circumstances. Then, if a disaster occurs, an alternate node can take ownership of applications and data, provided the replicated data is current and consistent.

As shown in Figure 1-8 on page 28, physical replication can be done in software or hardware.
MirrorDisk/UX is an example of physical replication done in the software; a disk I/O is written to each array connected to the node, requiring the node to make multiple disk I/Os. Continuous Access XP on the HP SureStore E Disk Array XP series is an example of physical replication in hardware; a single disk I/O is replicated across the Continuous Access link to a second XP disk array.

**Advantages of physical replication in hardware** are:

- There is little or no lag time writing to the replica. This means that data remains very current.

- Replication consumes no additional CPU.

- The hardware deals with resynchronization if the link or disk fails. And resynchronization is independent of CPU failure; if the CPU fails and the disk remains up, the disk knows it does not have to be resynchronized.

- Write mode is configurable and applies unless the disk or link is down.

- Data can be copied in both directions, so that if the primary fails and the replica takes over, data can be copied back to the primary when it comes back up.
Disadvantages of physical replication in hardware are:

- The logical order of data writes is not maintained. When a replication link goes down and transactions continue at the primary site, writes to the primary disk are queued in a bit-map. When the link is restored, if there has been more than one write to the primary disk, there is no way to determine the original order of transactions. This increases the risk of data inconsistency.

**NOTE**

Configuring the disk so that it does not allow a subsequent disk write until the current disk write is copied to the replica (synchronous writes) can limit this risk as long as the link remains up. Synchronous writes impact the capacity and performance of the data replication technology.

Also, because the replicated data is a write to a physical disk block, database corruption and human errors, such as the accidental removal of a database table, are replicated at the remote site.

- Redundant disk hardware and cabling are required. This at least doubles data storage costs. Also, because the technology is in the disk itself, this solution requires specialized hardware.

- Because the technology is in the disk itself, this solution requires specialized hardware.

- For architectures using dedicated cables, the distance between the sites is limited by the cable that connects the arrays. The Continuous Access XP and EMC SRDF technology are limited to 60 kilometers; FibreChannel is limited to 10 kilometers.

- For architectures using common carriers, the connections are less reliable and more expensive.

Advantages of physical replication in software are:

- There is little or no time lag between the initial and replicated disk I/O, so data remains very current.

- The solution is independent of disk technology, so you can use any disk technology you want.

- Data copies are peers, so there is no issue with reconfiguring a replica to function as a primary disk after failover.

- Because there are multiple read devices, that is, the node has access
Disaster Tolerance and Recovery in an MC/ServiceGuard Cluster

**Disaster Tolerant Architecture Guidelines**

- To both copies of data, there may be improvements in read performance.

- Writes are synchronous unless the link or disk is down.

**Disadvantages of physical replication in software** are:

- As with physical replication in the hardware, the logical order of data writes is not maintained. When the link is restored, if there has been more than one write to the primary disk, there is no way to determine the original order of transactions.

**NOTE**

Configuring the software so that a write to disk must be replicated on the remote disk before a subsequent write is allowed can limit the risk of data inconsistency while the link is up.

- Additional hardware is required for the cluster.

- Distance between sites is limited by the physical disk link capabilities: 25 meters for F/W SCSI, and 10 km for FibreChannel.

- Performance is affected by many factors: CPU overhead for mirroring, double I/Os, degraded write performance, and CPU time for resynchronization. In addition, CPU failure may cause a resynchronization even if it is not needed, further affecting system performance.

**Logical Data Replication** Logical data replication is a method of replicating data by repeating the sequence of transactions at the remote site. Logical replication often must be done at both the file system level, and the database level in order to replicate all of the data associated with an application. Most database vendors have one or more database replication products. An example is the Oracle Standby Database.

Logical replication can be configured to use synchronous or asynchronous writes. Transaction processing monitors (TPMs) can also perform logical replication.
Advantages of using logical replication are:

- The distance between nodes is limited only by the networking technology.
- There is no additional hardware needed to do logical replication, unless you choose to boost CPU power and network bandwidth.
- Logical replication can be implemented to reduce risk of duplicating human error. For example, if a database administrator erroneously removes a table from the database, a physical replication method will duplicate that error at the remote site as a raw write to disk. A logical replication method can be implemented to only replicate database transactions, not database commands, so such errors would not be replicated at the remote site. This also means that administrative tasks, such as adding or removing database tables, has to be repeated at each site.
- With database replication you can roll transactions forward or backward to achieve the desired level of currency on the replica, although this functionality is not available with file system replication.

Disadvantages of logical replication are:

- It uses significant CPU overhead because transactions are often replicated more than once and logged to ensure data consistency, and all but the most simple database transactions take significant CPU. It also uses network bandwidth, whereas most physical replication methods use a separate data replication link. As a result, there may be a significant lag in replicating transactions at the remote site, which affects data currency.
Disaster Tolerance and Recovery in an MC/ServiceGuard Cluster

Disaster Tolerant Architecture Guidelines

- If the primary database fails and is corrupt, and the replica takes over, the process for restoring the primary database so that it can be used as the replica is complex. It often involves recreating the database and doing a database dump from the replica.

- Applications often have to be modified to work in an environment that uses a logical replication database. Logic errors in applications or in the RDBMS code itself that cause database corruption will be replicated to remote sites. This is also an issue with physical replication.

Ideal Data Replication  The ideal disaster tolerant architecture, if budgets allow, is a combination of physical data replication for performance and data currency, backed up by logical data replication for data consistency. The logical replica would only be used in the cases where the physical replicas were corrupt.

Using Alternative Power Sources

In a high-availability cluster, redundancy is applied to cluster components, such as PV links, redundant network cards, power supplies, and disks. In disaster tolerant architectures another level of protection is required for these redundancies.

Each data center that houses part of a disaster tolerant cluster should be supplied with power from a different circuit. In addition to a standard UPS (uninterrupted power supply), each node in a disaster tolerant cluster should be on a separate power circuit; see Figure 1-10.

Figure 1-10  Alternative Power Sources
Housing remote nodes in another building often implies they are powered by a different circuit, so it is especially important to make sure all nodes are powered from a different source if the disaster tolerant cluster is located in two data centers in the same building. Some disaster tolerant designs go as far as making sure that their redundant power source is supplied by a different power substation on the grid. This adds protection against large-scale power failures, such as brown-outs, sabotage, or electrical storms.

**Creating Highly Available Networking**

Standard high-availability guidelines require redundant networks. Redundant networks may be highly available, but they are not disaster tolerant if a single accident can interrupt both network connections. For example, if you use the same trench to lay cables for both networks, you do not have a disaster tolerant architecture because a single accident, such as a backhoe digging in the wrong place, can sever both cables at once, making automated failover during a disaster impossible.

In a disaster tolerant architecture, the reliability of the network is paramount. To reduce the likelihood of a single accident causing both networks to fail, redundant network cables should be installed so that they use physically different routes for each network as indicated in Figure 1-11 on page 34. How you route cables will depend on the networking technology you use. Specific guidelines for some network technologies are listed here.
Disaster Tolerance and Recovery in an MC/ServiceGuard Cluster

Disaster Tolerant Architecture Guidelines

Disaster Tolerant Local Area Networking

The configurations described in this section are for FDDI and Ethernet based Local Area Networks.

Figure 1-11 Reliability of the Network is Paramount

Wrong:
Cables use same route.

Data Center A

Accident severs both network cables.
Disaster recovery impossible.

Data Center B

Right:
Cables use different route.

Data Center A

Accident severs one network cable.
Disaster recovery still possible.

Data Center B

If you use FDDI networking, you may want to use one of these configurations, or a combination of the two:

- Each node has two SAS (Single Attach Station) host adapters with redundant concentrators and a different dual FDDI ring connected to each adapter. As per disaster tolerant architecture guidelines, each ring must use a physically different route.
Disaster Tolerance and Recovery in an MC/ServiceGuard Cluster

Disaster Tolerant Architecture Guidelines

- Each node has two DAS (Dual Attach Station) host adapters, with each adapter connected to a different FDDI bypass switch. The optical bypass switch is used so that a node failure does not bring the entire FDDI ring down. Each switch is connected to a different FDDI ring, and the rings are installed along physically different routes.

**Figure 1-12** Highly Available FDDI Network: Two Options

Ethernet networks can also be used to connect nodes in a disaster tolerant architecture within the following guidelines:

- Each node is connected to redundant hubs and bridges using two 10BaseT or 100BaseT host adapters. Bridges, repeaters, or other components that convert from copper to fiber cable may be used to span longer distances.

- Redundant Ethernet links must be routed in the opposite directions, otherwise a data center failure can cause a failure of both networks.
Disaster Tolerance and Recovery in an MC/ServiceGuard Cluster

Disaster Tolerant Architecture Guidelines

Figure 1-13 on page 36 shows one Ethernet link routed from data center A, to B, to C. The other is routed from A, to C, to B. If both links were routed from A, to C, to B, the failure of data center C would make it impossible for clients to connect to data center B in the case of failover.

**Figure 1-13** Routing Highly Available Ethernet Connections in Opposite Directions

Disaster Tolerant Wide Area Networking

Disaster tolerant networking for continental clusters is directly tied to the data replication method. In addition to the redundant lines connecting the remote nodes, you also need to consider what bandwidth you need to support the data replication method you have chosen. A continental cluster that handles a high number of transactions per minute will not only require a highly available network, but also one with a large amount of bandwidth.

This is a brief discussion of things to consider when choosing the network...
Disaster Tolerance and Recovery in an MC/ServiceGuard Cluster

Disaster Tolerant Architecture Guidelines

configuration for your continental cluster. Details on WAN choices and configurations can be found in a white paper available from http://docs.hp.com/hpux/ha.

- Bandwidth affects the rate of data replication, and therefore the currency of the data should you need to switch control to another site. The greater the number of transactions you process, the more bandwidth you will need. The following connection types offer differing amounts of bandwidth:
  - T1 and T3 are low-end at 1Mbps and 10-20 Mbps respectively.
  - ISDN is next at 128Mbps; DSL offers an alternative to ISDN, but may not be readily available in some areas.
  - ATM provides 256Mbps but is best suited for transfer of audio and visual data.

- Reliability affects whether or not data replication happens, and therefore the consistency of the data should you need to fail over to the recovery cluster. Redundant leased lines should be used, and should be from two different common carriers, if possible.

- Cost influences both bandwidth and reliability. Higher bandwidth and dual leased lines cost more. It is best to address data consistency issues first by installing redundant lines, then weigh the price of data currency and select the line speed accordingly.

Disaster Tolerant Cluster Limitations

Disaster tolerant clusters have limitations, some of which can be mitigated by good planning. Some examples of MPOF that may not be covered by disaster tolerant configurations:

- Failure of all networks among all data centers — This can be mitigated by using a different route for all network cables.

- Loss of power in more than one data center — This can be mitigated by making sure data centers are on different power circuits, and redundant power supplies are on different circuits. If power outages are frequent in your area, and down time is expensive, you may want to invest in a backup generator.

- Loss of all copies of the on-line data — This can be mitigated by replicating data off-line (frequent backups). It can also be mitigated by taking snapshots of consistent data and storing it on-line.
Disaster Tolerance and Recovery in an MC/ServiceGuard Cluster

Disaster Tolerant Architecture Guidelines

Business Copy XP and EMC Symmetrix BCV (Business Consistency Volumes) provide this functionality and the additional benefit of quick recovery should anything happen to both copies of on-line data.

- A **rolling disaster** is a disaster that occurs before the cluster is able to recover from a non-disastrous failure. An example is a data replication link that fails, then, as it is being restored and data is being resynchronized, a disaster causes an entire data center to fail. The effects of rolling disasters can be mitigated by ensuring that a copy of the data is stored either off-line or on a separate disk that can quickly be mounted. The trade-off is a lack of currency for the data in the off-line copy.
Managing a Disaster Tolerant Environment

In addition to the changes in hardware and software to create a disaster tolerant architecture, there are also changes in the way you manage the environment. Configuration of a disaster tolerant architecture needs to be carefully planned, implemented and maintained. There are additional resources needed, and additional decisions to make concerning the maintenance of a disaster tolerant architecture:

• Manage it in-house, or hire a service?

Hiring a service can remove the burden of maintaining the capital equipment needed to recover from a disaster. Most disaster recovery services provide their own off-site equipment, which reduces maintenance costs. Often the disaster recovery site and equipment are shared by many companies, further reducing cost.

Managing disaster recovery in-house gives complete control over the type of redundant equipment used and the methods used to recover from disaster, giving you complete control over all means of recovery.

• Implement automated or manual recovery?

Manual recovery costs less to implement and gives more flexibility in making decisions while recovering from a disaster. Evaluating the data and making decisions can add to recovery time, but it is justified in some situations, for example if applications compete for resources following a disaster and one of them has to be halted.

Automated recovery reduces the amount of time and in most cases eliminates human intervention needed to recover from a disaster. You may want to automate recovery for any number of reasons:

— Automated recovery is usually faster.
— Staff may not be available for manual recovery, as is the case with “lights-out” data centers.
— Reduction in human intervention is also a reduction in human error. Disasters don’t happen often, so lack of practice and the stressfulness of the situation may increase the potential for human error.
— Automated recovery can be transparent to the clients.
Disaster Tolerance and Recovery in an MC/ServiceGuard Cluster

Managing a Disaster Tolerant Environment

Even if recovery is automated, you may choose to, or need to recover from some types of disasters with manual recovery. A rolling disaster, which is a disaster that happens before the cluster has recovered from a previous disaster, is an example of when you may want to manually switch over. If the data link failed, and as it was coming up and resynchronizing data, a data center failed, you would want human intervention to make judgment calls on which site had the most current and consistent data before failing over.

• Who manages the nodes in the cluster and how are they trained?

Putting a disaster tolerant architecture in place without planning for the people aspects is a waste of money. Training and documentation are more complex because the cluster is in multiple data centers.

Each data center often has its own operations staff with their own processes and ways of working. These operations people will now be required to communicate with each other and coordinate maintenance and failover rehearsals, as well as working together to recover from an actual disaster. If the remote nodes are placed in a “lights-out” data center, the operations staff may want to put additional processes or monitoring software in place to maintain the nodes in the remote location.

Rehearsals of failover scenarios are important to keep everyone prepared. A written plan should outline rehearsal of what to do in cases of disaster with a minimum recommended rehearsal schedule of once every 6 months, ideally once every 3 months.

• How is the cluster maintained?

Planned downtime and maintenance, such as backups or upgrades, must be more carefully thought out because they may leave the cluster vulnerable to another failure. For example, in the MC/ServiceGuard configurations discussed in Chapter 2, nodes need to be brought down for maintenance in pairs: one node at each site, so that quorum calculations do not prevent automated recovery if a disaster occurs during planned maintenance.

Rapid detection of failures and rapid repair of hardware is essential so that the cluster is not vulnerable to additional failures.

Testing is more complex and requires personnel in each of the data centers. Site failure testing should be added to the current cluster testing plans.
Using this Guide with Your Disaster Tolerant Cluster Products

The rest of this guide provides detailed, task-oriented documentation covering a variety of HP products and technologies, as follows:

- Chapter 2, “Building a Campus Cluster Using FibreChannel and MC/ServiceGuard,” shows different types of FibreChannel configuration for a campus cluster using standard MC/ServiceGuard (B3935BA).

- Chapter 3, “Building a Metropolitan Cluster Using MetroCluster with Continuous Access XP,” describes the metropolitan cluster architecture as implemented with the HP SureStore E Disk Array XP Series and the Continuous Access XP software. This chapter is intended for users of MetroCluster with Continuous Access XP (B8109BA).

- Chapter 4, “Building a Metropolitan Cluster Using MetroCluster with EMC SRDF,” describes the metropolitan cluster architecture as implemented with the EMC Symmetrix disk arrays and the EMC SRDF facility. This chapter is intended for users of MetroCluster with EMC SRDF (B6264BA).

- Chapter 5, “Building a Continental Cluster,” presents the steps for creating a wide area configuration using the ContinentalClusters product together with various data replication products, including Continuous Access XP and EMC SRDF. This chapter is intended for users of ContinentalClusters (B7659BA).

- Chapter 6, “Physical Data Replication for ContinentalClusters Using Continuous Access XP,” is intended for users of physical data replication with the HP SureStore E Disk Array XP Series in a continental cluster.

- Chapter 7, “Physical Data Replication for ContinentalClusters Using EMC SRDF,” is intended for users of physical data replication with the EMC Symmetrix disk array and EMC SRDF in a continental cluster.

- Chapter 8, “Cascading Failover in a Continental Cluster,” describes a specific configuration using the EMC Symmetrix disk array to provide data replication with disaster recovery between a continental
Disaster Tolerance and Recovery in an MC/ServiceGuard Cluster

Using this Guide with Your Disaster Tolerant Cluster Products

cluster in one geographic region and an ordinary ServiceGuard cluster in another region.

A set of appendixes and a glossary provide additional reference information.
Building a Campus Cluster Using FibreChannel and MC/ServiceGuard

An MC/ServiceGuard cluster can be configured as a disaster tolerant campus cluster using FibreChannel technology to connect nodes to mass storage devices.

This chapter assumes that readers are already familiar with MC/ServiceGuard configuration tasks, MC/ServiceGuard campus cluster concepts and installation procedures, and FibreChannel host adapters.

The topics discussed in this chapter are:

• Guidelines for Disaster Tolerant Architectures with FibreChannel
• Two Data Centers with FibreChannel Hubs
• Two Data Centers with FibreChannel Point-to-Point
• Advantages and Disadvantages of a Two-Data-Center Architecture
Guidelines for Disaster Tolerant Architectures with FibreChannel

FibreChannel host adapters make it possible to increase the distance between nodes in an MC/ServiceGuard cluster thus making it possible to design a disaster tolerant architecture for campus or metropolitan clusters. The following configurations are supported:

- Two data centers using FibreChannel hubs. This architecture can be implemented using a variety of disk arrays. Consult the HP 9000 Servers Configuration Guide (available through your HP representative) for a list of supported HA disks.

- Two data centers using FibreChannel point-to-point and HP SureStore E Disk Array XP Series or EMC Symmetrix disk arrays.

The two-data-center architecture is based on a standard MC/ServiceGuard configuration with half of the nodes in one data center, and the other half in another data center. Nodes can be located in separate data centers in the same building, or even separate buildings within the limits of FibreChannel technology. FibreChannel has a maximum distance of 500 meters (10 kilometers with long-wave hubs). F/W (Fast/Wide) SCSI only allows a maximum length of 25 meters.

Following are the disaster tolerant architecture requirements for campus clusters:

- Data centers must be self-contained such that the loss of one data center does not cause the entire cluster to fail. All Single Points of Failure (SPOFs) must be eliminated so that surviving systems continue to run if one or more systems fail.

- The number of nodes in each data center must be equal.

- Heartbeat networks between the data centers must be redundant and routed along physically different paths so that the loss of any one data center does not cause the network between surviving data centers to fail. All nodes in the cluster must be connected to the redundant heartbeat network links.

If you are using FibreChannel technology for the network as well, you must use completely different hardware from that used to connect nodes to mass storage devices. This includes cables, repeaters, hubs,
Building a Campus Cluster Using FibreChannel and MC/ServiceGuard
Guidelines for Disaster Tolerant Architectures with FibreChannel

and anything else related to the FibreChannel fabric.

• Data must be mirrored at the remote site using a product such as MirrorDisk/UX. Data connections must be redundant and routed differently, like heartbeat network cables are. It is recommended to route FibreChannel cables and heartbeat network cables in such a way that no two failures can disable all heartbeat networks and both FibreChannel connections.

• Data components, e.g. FibreChannel hubs, must be redundant and powered using alternative power sources as described in Chapter 1. All failover nodes must be connected to both mirror copies of the data.

• Dual cluster lock disks are required, one in each data center. All nodes must be connected to both cluster lock disks. Each cluster lock disk must be on a different FibreChannel arbitrated loop.
Two Data Centers with FibreChannel Hubs

Although this architecture should work with the maximum number of nodes allowed in an MC/ServiceGuard cluster, this cluster architecture has been tested with a maximum of 4 nodes. Therefore, the configuration currently supported is 2 nodes per data center as shown in Figure 2-1.

This configuration can be implemented using any HP-supported FibreChannel devices. Disks must be available from all nodes using redundant links. Not all links are shown in Figure 2-1. The two cluster lock disks should be located on separate FibreChannel loops to guard against single point of failure.

The lock disks can also be used as data disks. They must be connected to all nodes using redundant links. Not all links are shown in Figure 2-1. If you do choose to use High Availability Model 30 disk arrays as cluster lock disks, you must disable the AutoTrespass function for all LUNs, which prevents you from using PV links thus introducing a single point of failure.

Nodes can connect to disks in the same data center using short wave
Building a Campus Cluster Using FibreChannel and MC/ServiceGuard

Two Data Centers with FibreChannel Hubs

ports, and hubs can connect between data centers using long-wave ports. This gives you a maximum distance of 3 kilometers between data centers, making it possible to locate data centers in different buildings. Therefore this configuration can be used to create either campus or metropolitan clusters.

This architecture is also possible using FibreChannel switches.
Two Data Centers with FibreChannel Point-to-Point

Although this architecture should work with the maximum number of nodes allowed in an MC/ServiceGuard cluster, this cluster architecture has been tested with a maximum of 4 nodes and dual lock disks. Therefore, the configuration currently supported is 2 nodes per data center as shown in Figure 2-2.

Disks in the configuration can be HP SureStore E XP Series disk arrays or EMC Symmetrix arrays. Each array must contain one of the dual cluster lock disks. Host connections to the disks must be made using redundant links (PV links).

The maximum distance between nodes is 10 kilometers using long-wave hubs between the disk arrays. Therefore, this configuration can be used for both campus clusters and metropolitan clusters.
Advantages and Disadvantages of a Two-Data-Center Architecture

The advantages of a two-data-center architecture are:

- Lower cost.
- Only two data centers are needed, meaning less space and less coordination between operations staff.
- No arbitrator nodes are needed.
- All systems are connected to both copies of data, so that if a primary disk fails, but the primary system stays up, there is no package failover.

The disadvantages of a two-data-center architecture are:

- There is a slight chance of split brain syndrome. Because there are two cluster lock disks, you would get split brain syndrome if the following occurred simultaneously:
  - All heartbeat networks fail.
  - All disk links fail. The disk link from data center A to cluster lock disk B fails (see Figure 2-1 on page 46.) And the disk link from data center B to cluster lock disk A fails.

The chances are slight, however these events happening at the same time would result in split brain syndrome and probable data inconsistency. Planning different physical routes for both network and data connections or adequately protecting the physical routes greatly reduces the possibility of split brain syndrome.

- Currently there is a maximum distance of 10 kilometers between data centers.
- Mirroring increases CPU overhead.
- Although it is a low cost solution, it does require some additional cost:
  - FibreChannel links are required for both local and remote connectivity.
  - All systems must be connected to both copies of the data.
Building a Campus Cluster Using FibreChannel and MC/ServiceGuard

Advantages and Disadvantages of a Two-Data-Center Architecture
3

Building a Metropolitan Cluster Using MetroCluster/CA

This chapter describes the high availability toolkit template for MetroCluster with Continuous Access XP (hereafter known as MetroCluster/CA), which allows you to install and configure an MC/ServiceGuard metropolitan cluster using the HP SureStore E Disk Array XP Series with Continuous Access XP. It is assumed that readers are already familiar with MC/ServiceGuard configuration tasks, MC/ServiceGuard metropolitan cluster concepts and installation procedures, HP SureStore E Disk Array XP Series and Continuous Access XP concepts, and Raid Manager configuration and use.

The topics discussed in this chapter are:

- Overview of MetroCluster/CA
- Designing a Disaster Tolerant Architecture for use with MetroCluster/CA
- Preparing an MC/ServiceGuard Cluster for MetroCluster/CA
- Configuring Packages for Automatic Disaster Recovery
- Maintaining a Cluster that uses MetroCluster/CA
Overview of MetroCluster/CA

MetroCluster is a set of scripts that work in an MC/ServiceGuard cluster to automate failover to alternate nodes in the case of disaster in a campus or metropolitan cluster. The MetroCluster/CA template contains the following files, installed in `/opt/cmcluster/toolkit/SGCA`:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sgcapkg.cntl</td>
<td>The MetroCluster/CA control script template. This template must be customized for the specific Disk Array XP Series and HP 9000 host system configuration. Copies of this template must be customized for each separate MC/ServiceGuard package.</td>
</tr>
<tr>
<td>Samples</td>
<td>A directory containing sample convenience shell scripts that must be edited before using. These shell scripts may help to automate some configuration tasks. These scripts are contributed, and not supported. This directory also contains sample Raid Manager configuration files and other examples.</td>
</tr>
</tbody>
</table>

MetroCluster/CA has to be installed on all nodes that will run a ServiceGuard package whose data are on an HP SureStore E Disk Array XP Series where the data are replicated to a second XP using the Continuous Access XP facility. In the event of node failure, the integration of MetroCluster/CA with the package will allow the application to fail over in the following ways:

- among local host systems attached to the same XP Series array
- between one system that is attached locally to its XP and another “remote” host that is attached locally to the other XP

Configuration of MetroCluster/CA must be done on all the cluster nodes, as is done for any other MC/ServiceGuard package. To use MetroCluster/CA, Raid Manager XP host-based software for control and status of the XP Series boxes must also be installed and configured on each HP 9000 host system that might execute the application package.
Designing a Disaster Tolerant Architecture for use with MetroCluster/CA

MetroCluster/CA is designed for use in a campus cluster or metropolitan cluster environment within the 100 km limit of the FDDI network.

All nodes must be members of a single MC/ServiceGuard cluster. Two configurations are supported:

• A single data center without arbitrators (not disaster tolerant.)
• A three data center architecture with one or two arbitrator systems. See Figure 3-1 on page 54.

Following are the disaster tolerant architecture requirements:

• In the disaster tolerant cluster architecture, it is expected that each data center is self-contained such that the loss of one data center does not cause the entire cluster to fail. It is important that all single points of failure (SPOF) be eliminated so that surviving systems continue to run in the event that one or more systems fail.

• It is also expected that the networks between the data centers are redundant and routed in such a way that the loss of any one data center does not cause the network between surviving data centers to fail.

• Exclusive volume group activation must be used for all VGs associated with packages that use the XP Series disk array. The design of the MetroCluster/CA script assumes that only one system in the cluster will have a VG activated at any time.

Single Data Center

A single data center architecture is supported, but it is not a true disaster tolerant architecture. If the entire data center fails, there will be no automated failover. This architecture is only valid for protecting data through data replication, and for protecting against multiple node failures.

Three Data Centers with Arbitrator(s)

This is the recommended and supported disaster tolerant architecture
Building a Metropolitan Cluster Using MetroCluster/CA

Designing a Disaster Tolerant Architecture for use with MetroCluster/CA

for use with MetroCluster/CA. The three data center architecture consists of two data centers with an equal number of nodes and a third data center with one or more arbitrator nodes; see Figure 3-1.

Figure 3-1 Three Data Centers with Arbitrators

The local XP Series disk array is called the **Main Control Unit** for all nodes and packages in a given data center. The remote XP Series disk array, where the data is replicated, is called the **Remote Control Unit**. Note that main and remote are relative terms. An XP Series disk array can be the main disk array for one set of packages and the remote disk array for another. In Figure 3-1, the XP disk array in data center A is the main or primary disk array for packages A and B, and the remote or secondary disk array for packages C and D in data center B. For packages A and B, data is written to PVOLs on the the array in Data Center A.
Building a Metropolitan Cluster Using MetroCluster/CA

Designing a Disaster Tolerant Architecture for use with MetroCluster/CA

Center A and replicated to SVOLs on the array in Data Center B. Likewise the XP disk array in Data Center B is the primary or main disk array for packages C and D, and the secondary or remote for packages A and B. For packages C and D, data is written to PVOLs on the disk array in Data Center B and replicated to SVOLs in Data Center A.

Arbitrators provide functionality like that of the cluster lock disk, and act as tie-breakers for a cluster quorum in case all of the nodes in one data center go down at the same time. Cluster lock devices are not used in the three-data-center architecture because cluster locks cannot be maintained across the ESCON link.

Arbitrators are fully functioning systems that are members of the cluster, and are not usually physically connected to the XP disk arrays. Table 3-2 lists the allowable number of nodes at each data center in a three data center configuration, up to a 16-node maximum cluster size. (Note that the maximum cluster size for MC/ServiceGuard A.10.10 and A.10.11 is 8 nodes.)

Table 3-2  
Supported System and Data Center Combinations

<table>
<thead>
<tr>
<th>Data Center A</th>
<th>Data Center B</th>
<th>Data Center C (Arbitrators)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2*</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>2*</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>2*</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>2*</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>2*</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>2*</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>2*</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>2*</td>
</tr>
</tbody>
</table>
Building a Metropolitan Cluster Using MetroCluster/CA

Designing a Disaster Tolerant Architecture for use with MetroCluster/CA

* Configurations with two arbitrators are preferred because they provide a greater degree of availability, especially in cases when a node is down due to a failure or planned maintenance.

NOTE
In the campus or metropolitan environment, the same number of systems must be present in each of the two data centers (Data Center A and Data Center B) whose systems are connected to the XP disk arrays. There must be either one or two arbitrators in Data Center C.

Arbitrator Node Configuration Rules
Although you can use one arbitrator, having two arbitrators provides greater flexibility in taking systems down for planned outages as well as providing better protection against multiple points of failure. Using two arbitrators:

• Provides local failover capability to applications running on the arbitrator.
• Protects against multiple points of failure (MPOF).
• Provides for planned downtime on a single system anywhere in the cluster.

If you use a single arbitrator system, special procedures must be followed during planned downtime to remain protected. Systems must be taken down in pairs, one from each of the data centers, so that the MC/ServiceGuard quorum is maintained after a node failure. If the arbitrator itself must be taken down, disaster recovery capability is at risk if one of the other systems fails.

Arbitrator systems can be used to perform important and useful work such as:

• Hosting mission-critical applications not protected by disaster recovery software
• Running monitoring and management tools such as IT/Operations or Network Node Manager
• Running backup applications such as Omniback
• Acting as application servers
**Chapter 3**

**Building a Metropolitan Cluster Using MetroCluster/CA**

**Designing a Disaster Tolerant Architecture for use with MetroCluster/CA**

**XP Series Disk Array Configuration Rules**

Each XP Series disk array must be configured with redundant ESCON links, each of which is connected to a different LCP or RCP card. To prevent a single point of failure (SPOF), there must be at least two physical boards in each XP for the ESCON links. Each board usually has multiple ports. However, a redundant ESCON link must be connected to a port on a different physical board from the board that has the primary ESCON link. When using bi-directional configurations, where data center A backs up data center B and data center B backs up data center A, you must have at least four ESCON links, two in each direction. Four ESCON links are also required in uni-directional configurations in which you want to allow failback.

**Calculating a Cluster Quorum**

When a cluster initially forms, all systems must be available to form the cluster (100% Quorum requirement).

A quorum is dynamic and is recomputed after each system failure. For instance, if you start out with an 8-node cluster and two systems fail, that leaves 6 out 8 surviving nodes, or a 75% quorum. The cluster size is reset to 6 nodes. If two more nodes fail, leaving 4 out of 6, quorum is 67%.

Each time a cluster forms, there must be more than 50% quorum to reform the cluster. A cluster lock disk is normally used as the tie-breaker when quorum is exactly 50%. However, a cluster lock disk is not supported with MetroCluster with Continuous Access XP. Therefore, a quorum of 50% or less will cause the remaining nodes to halt.

**Example Failover Scenarios with One Arbitrator**

Taking a node off-line for planned maintenance is treated the same as a node failure in these scenarios. Study these scenarios to make sure you do not put your cluster at risk during planned maintenance.
Designing a Disaster Tolerant Architecture for use with MetroCluster/CA

Figure 3-2  Failover Scenario with a Single Arbitrator

The scenarios in Table 3-3, based on Figure 3-2, illustrate possible results if one or more nodes fail in a configuration with a single arbitrator.

Table 3-3  Node Failure Scenarios with One Arbitrator

<table>
<thead>
<tr>
<th>Failure</th>
<th>Quorum</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>arbitrator 1</td>
<td>4 of 5 (80%)</td>
<td>no change</td>
</tr>
<tr>
<td>node 1</td>
<td>4 of 5 (80%)</td>
<td>pkg A switches</td>
</tr>
<tr>
<td>node 1, then node 2</td>
<td>3 of 4 (75%)</td>
<td>pkg A and B switch</td>
</tr>
<tr>
<td>node 1, 2, then arbitrator 1</td>
<td>2 of 3 (67%)</td>
<td>pkg A and B switch</td>
</tr>
<tr>
<td>nodes 1, 2, arbitrator 1, then node 3</td>
<td>1 of 2 (50%)</td>
<td>cluster halts*</td>
</tr>
<tr>
<td>arbitrator 1, then node 1</td>
<td>3 of 4 (75%)</td>
<td>pkg A switches</td>
</tr>
</tbody>
</table>

* Cluster can be manually started with the remaining node.
Building a Metropolitan Cluster Using MetroCluster/CA

Designing a Disaster Tolerant Architecture for use with MetroCluster/CA

Table 3-4 illustrates possible results if a data center fails in a configuration with a single arbitrator.

**Table 3-4  Data Center Failure Scenarios with One Arbitrator**

<table>
<thead>
<tr>
<th>Failure</th>
<th>Quorum</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>data center A (nodes 1 and 2)</td>
<td>3 of 5 (60%)</td>
<td>pkg A and B switch to data center B</td>
</tr>
<tr>
<td>data center A, then arbitrator 1</td>
<td>2 of 3 (67%)</td>
<td>pkg A and B switch, then no change</td>
</tr>
<tr>
<td>data center A and arbitrator 1</td>
<td>2 of 5 (40%)</td>
<td>cluster halts*</td>
</tr>
<tr>
<td>data center A, then arbitrator 1, then node 3</td>
<td>1 of 2 (50%)</td>
<td>cluster halts*</td>
</tr>
<tr>
<td>arbitrator 1, then data center A</td>
<td>2 of 4 (50%)</td>
<td>cluster halts*</td>
</tr>
<tr>
<td>node 3, then data center A</td>
<td>2 of 4 (50%)</td>
<td>cluster halts*</td>
</tr>
<tr>
<td>data center B</td>
<td>3 of 5 (60%)</td>
<td>pkg C and D switch to data center A</td>
</tr>
<tr>
<td>data center C</td>
<td>4 of 5 (80%)</td>
<td>no change</td>
</tr>
</tbody>
</table>

* Cluster can be manually started with the remaining node.

With a single arbitrator node, the cluster is at risk each time a node fails or comes down for planned maintenance.

**Example Failover Scenarios with Two Arbitrators**

Having two arbitrator nodes adds extra protection during node failures and allows you to do planned maintenance on arbitrator nodes without losing the cluster should a disaster occur.
Building a Metropolitan Cluster Using MetroCluster/CA
Designing a Disaster Tolerant Architecture for use with MetroCluster/CA

**Figure 3-3**  Failover Scenario with Two Arbitrators

![Diagram of failover scenario with two arbitrators](image)

The scenarios in Table 3-5 illustrate possible results if a data center or one or more nodes fail in a configuration with two arbitrators. Note that 3 of the 4 scenarios that caused a cluster halt with a single arbitrator, do not cause a cluster halt with two arbitrators.

**Table 3-5**  Failure Scenarios with Two Arbitrators

<table>
<thead>
<tr>
<th>Failure</th>
<th>Quorum</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>data center A (nodes 1 and 2)</td>
<td>4 of 6 (67%)</td>
<td>pkg A and B switch to data center B</td>
</tr>
<tr>
<td>data center A, then arbitrator 1</td>
<td>3 of 4 (75%)</td>
<td>pkg A and B switch, then no change</td>
</tr>
<tr>
<td>data center A and arbitrator 1</td>
<td>3 of 6 (50%)</td>
<td>cluster halts*</td>
</tr>
<tr>
<td>data center A, then arbitrator 1, then node 3</td>
<td>2 of 3 (67%)</td>
<td>pkg A, B, and C switch</td>
</tr>
<tr>
<td>arbitrator 1, then data center A</td>
<td>3 of 5 (60%)</td>
<td>pkg A and B switch to data center B</td>
</tr>
</tbody>
</table>
Building a Metropolitan Cluster Using MetroCluster/CA

Designing a Disaster Tolerant Architecture for use with MetroCluster/CA

Table 3-5  

<table>
<thead>
<tr>
<th>Failure</th>
<th>Quorum</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>node 3, then data center A</td>
<td>3 of 5 (60%)</td>
<td>pkg A and B switch to data center B</td>
</tr>
<tr>
<td>data center B</td>
<td>4 of 6 (67%)</td>
<td>pkg C and D switch to data center A</td>
</tr>
<tr>
<td>data center C</td>
<td>4 of 6 (67%)</td>
<td>no change</td>
</tr>
</tbody>
</table>

* Cluster can be manually started with the remaining node.

Disaster Tolerant Checklist

Use the this checklist to make sure you have adhered to the disaster tolerant architecture guidelines for a three-data-center configuration.

Figure 3-4  

Disaster Tolerant Checklist

- Data centers A and B have the same number of nodes to maintain quorum in case an entire data center fails.
- Arbitrator nodes are located in a separate data center from either of the primary data centers (A or B).
- The elements in each data center including nodes, disks, network components, and climate control are on separate power circuits.
- Each node is configured with PV links.
- Each XP Series disk array is configured with redundant ESCON links, each of which is connected to a different LCP or RCP card.
- At least two networks are configured to function as the cluster heartbeat.
- All redundant cabling for network, heartbeat, and ESCON connections are routed using different physical paths.
- Redundant XP Series command devices have been configured.
Cluster Configuration Worksheet

Use this cluster configuration worksheet either in place of, or in addition to the worksheet provided in the Managing MC/ServiceGuard manual. If you have already completed an MC/ServiceGuard cluster configuration worksheet, you only need to complete the first part of this worksheet.

### Name and Nodes

**Cluster Name:**

**Data Center A Name and Location:**

**Node Names:**

**Data Center B Name and Location:**

**Node Names:**

**Arbitrator Data Center C Name and Location:**

**Arbitrator Node Names:**

**Maximum Configured Packages:**

### Subnets

**Heartbeat IP Addresses:**

**Non-Heartbeat IP Addresses:**

### Timing Parameters

**Heartbeat Interval:**

**Node Time-out:**

**Network Polling Interval:**

**AutoStart Delay:**
Package Configuration Worksheet

Use this package configuration worksheet either in place of, or in addition to the worksheet provided in the Managing MC/ServiceGuard manual. If you have already completed an MC/ServiceGuard package configuration worksheet, you only need to complete the first part of this worksheet.

Figure 3-6

Package Configuration Worksheet

Package Configuration File Data

Package Name: ____________________________  Data Center: ____________________________
Primary Node: ____________________________  Data Center: ____________________________
First Failover Node: _______________________  Data Center: ____________________________
Second Failover Node: _____________________  Data Center: ____________________________
Third Failover Node: _______________________  Data Center: ____________________________
Fourth Failover Node: _____________________  Data Center: ____________________________
Package Run Script: ____________________________  Timeout: ____________
Package Halt Script: ____________________________  Timeout: ____________
Maximum Configured Packages: ____________

XP Series Volume Configuration

<table>
<thead>
<tr>
<th>Device Group</th>
<th>Device Name</th>
<th>Port #</th>
<th>Target ID</th>
<th>LUN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Package Control Script Data

VG[0]: _______ LV[0]: _______ FS[0]: _______ FS_MOUNT_OPT[0]: _______
VG[1]: _______ LV[1]: _______ FS[1]: _______ FS_MOUNT_OPT[1]: _______
VG[2]: _______ LV[2]: _______ FS[2]: _______ FS_MOUNT_OPT[2]: _______
IP[0]: _______ SUBNET[0]: _______
IP[1]: _______ SUBNET[1]: _______
X.25 Resource Name: ____________________________
Service Name: ____________ Run Command: ____________ Retries: __ RetryTime: __
Service Fail Fast Enabled?: ____________ Service Halt Timeout: ____________
Service Name: ____________ Run Command: ____________ Retries: __ RetryTime: __
Service Fail Fast Enabled?: ____________ Service Halt Timeout: ____________
Preparing an MC/ServiceGuard Cluster for MetroCluster /CA

When the following procedures are completed, an adoptive node will be able to access the data belonging to a package after it fails over.

**Setting up the Hardware**

Ensure that the XP Series disk arrays are correctly cabled using PV links to each node in the cluster that will run packages accessing data on the array.

- Configure the XP disk array for synchronous or asynchronous operation. If you are using a fence level of ASYNC, you must configure a **side file** using the Service Processor (SVP) attached to the XP system. Synchronous operation does not require side file configuration on the SVP.

- Use the `ioscan` command to determine what devices on the XP disk array have been configured as command devices. The device-specific information in the rightmost column of the `ioscan` output will have the suffix `-CM` for these devices, for example, OPEN-3-CM.

  If there are no configured command devices on the disk array, you must create two before proceeding. Each command device must have alternate links (PV links). The first command device is the primary command device. The second command device is a redundant command device and is used only upon failure of the primary command device. The command devices must be mapped to the various host interfaces by using the SVP (disk array console) or a remote console.

- Primary (PVOL) and secondary (SVOL) volumes must be correctly defined and assigned to the appropriate nodes in the XP hardware configuration.
Preparing an MC/ServiceGuard Cluster for MetroCluster /CA

Figure 3-7  XP Series Primary and Secondary Volume Definitions

- Primary devices (PVOLs) must be locally protected (RAID 1 or RAID 5).
- Secondary devices (SVOLs) must be locally protected (RAID 1 or RAID 5).

Setting Fence Levels

All devices defined in a given device group must be configured with the same fence level. A fence level of DATA or NEVER results in synchronous data replication; a fence level of ASYNC is used to enable asynchronous data replication.

Fence Level of DATA

Fence level = DATA is recommended to ensure a consistent copy of the data on all sides. If Fence level = DATA is not enabled, the data may be inconsistent in the case of a rolling disaster—additional failures taking place before the system has completely recovered from a previous failure.

Fence level = DATA is recommended to ensure that there is no possibility
of inconsistent data at the SVOL side in case of CA (ESCON) link failure. Since only dedicated ESCON links are supported, the probability of intermittent link failure is extremely low. Therefore, the probability of inconsistent data at the remote (SVOL) side is extremely low. However, inconsistent and therefore unusable data will result from the following sequence of circumstances:

- Fence level = DATA is not enabled.
- The CA links fail.
- The application continues to modify data.
- The link is restored.
- Resynchronization from PVOL to SVOL starts, but does not finish.
- The PVOL side fails

Although the risk of this sequence of events taking place is extremely low, if your business cannot afford even this quite small risk, then you must enable Fence level = DATA to ensure that the data at the SVOL side are always consistent.

The disadvantage of enabling Fence level = DATA is that when the CA link fails, or if the entire remote (SVOL) data center fails, all I/Os will be refused (to those devices) until the CA link is restored, or manual intervention is undertaken to split the PVOL from the SVOL. Applications may fail or may continuously retry the I/Os (depending on the application) if Fence level = DATA is enabled and the CA link fails.

**Fence Level of ASYNC**

Fence level = ASYNC is recommended to improve performance in data replication between the primary and the remote site. The XP disk array supports asynchronous mode with guaranteed ordering. When the host does a write I/O to the XP disk array, as soon as the data is written to cache, the array sends a reply to the host. A copy of the data with a sequence number is saved in an internal buffer, known as the side file, for later transmission to the remote XP disk array. When synchronous replication is used, the primary system cannot complete a transaction until a message is received acknowledging that data has been written to the remote site. With asynchronous replication, the transaction is completed once the data is written to the side file on the primary system, which allows I/O activity to continue even if the ESCON link is temporarily unavailable.
Building a Metropolitan Cluster Using MetroCluster/CA

Preparing an MC/ServiceGuard Cluster for MetroCluster /CA

The side file is 30% to 70% of cache (default 50%) that is assigned through the XP system's Service Processor (SVP). The **high water mark** (HWM) is 30% of the cache; if the quantity of data in the side file exceeds this value, the write I/O will be delayed to the side file starting from .5 seconds and increasing to 4 seconds maximum with every 5% increase over HWM in 500 ms increments.

If the HWM continues to grow, it will eventually hit the **side file threshold** (30 to 70% of cache). When this limit has been reached, the XP on the primary site cannot write to the XP on the secondary site until there is enough room in the side file. Before continuing to write, the primary XP will wait until there is enough room in the side file, and will keep trying until it reaches its **side file timeout** value, which is configured through the SVP. If timeout has been reached, then the primary XP disk array will begin tracking data on its bitmap which will be copied over to the secondary volume during resync.

The side file operation is shown in Figure 3-8.

**Figure 3-8** XP Series Disk Array Side File

![Diagram of XP Series Disk Array Side File]

**NOTE**

The side file must be configured using the XP Service Processor (SVP). Refer to the XP Series documentation for details.

**ESCON Link Timeout**

In asynchronous mode, when there is an ESCON link failure, both the PVOL and SVOL sides change to a PSUE state. When the SVOL side detects missing data blocks from the PVOL side, it will wait for those data blocks from the PVOL side until it has reached the configured ESCON link timeout value (set in the SVP). Once this timeout value has been reached, then the SVOL side will change to a PSUE state. The
default ESCON link timeout value is 5 minutes (300 seconds).

**Consistency Groups**

An important property of asynchronous mode volumes is the **consistency group** (CT group). A CT group is a grouping of LUNs that need to be treated the same from the perspective of data consistency (I/O ordering). A CT group can contain one or more device groups in the Raid Manager configuration file. A consistency group ID (CTGID), 0 to 15 for the XP256 or 0 to 63 for the XP512, is assigned automatically during pair creation.

**Limitations of Asynchronous Mode**

The following are restrictions for an asynchronous CT group in a Raid Manager configuration file:

- Asynchronous device groups cannot be defined to extend across multiple XP Series disk arrays.
- If two or more device groups are included within a CT group in the configuration file, then a pair operation occurs only at the granularity of the entire CT group.
- When making paired volumes, the Raid Manager registers a CTGID to the XP Series disk array automatically at `paircreate` time, and the device group in the configuration file is mapped to a CTGID. The maximum number of consistency groups per XP256 is 16 (0 to 15), and per XP512 is 64 (0 to 63). Efforts to create a CTGID with a higher number will be terminated with a return value of `EX_ENOCTG`.
- MetroCluster/CA will support only one package device group per consistency group. This means that in one metropolitan cluster, there can be only 16 packages that can be configured to use consistency groups on the XP256, or 32 packages that can be configured to use consistency groups on the XP512. MC/ServiceGuard supports only a maximum of 30 packages per cluster.

**Other Considerations on Asynchronous Mode**

The following are some additional considerations when using asynchronous mode:

- When adding a new volume to an existing device group, the new volume state is `SMPL`. The XP disk array controller (DKC) is smart enough to do the `paircreate` only on the new volume. If the device
Building a Metropolitan Cluster Using MetroCluster/CA

Preparing an MC/ServiceGuard Cluster for MetroCluster /CA

If a group has mixed volume states like PAIR and SMPL, the pairvolchk returns EX_ENQVOL, and horctakeover will fail.

- If you change the LDEV number associated with a given target/LUN, you must restart all the Raid Manager instances even though the Raid Manager configuration file is not modified.
- Any firmware update, cache expansion, or board change, requires a restart of all Raid Manager instances.
- pairsplit for asynchronous mode may take a long time depending on how long the synchronization takes. There is a potential for the link to fail while pairsplit is in progress. If this happens, pairsplit will fail with a return code of EX_EWSUSE.
- In most cases, MetroCluster/CA in asynchronous mode will behave the same as when the fence level is set to NEVER in synchronous mode.

Installing the Necessary Software

Before any configuration can begin, you need to perform the following installation tasks on all nodes:

- Install Raid Manager XP, which allows you to manage the XP series disk arrays from the node. Refer to the installation instructions in the Raid Manager XP User’s Guide.
- Edit the /etc/services file, adding an entry for the Raid Manager XP instance to be used with MetroCluster/CA in the format horcm<instance-number> <port-number>/udp. For example:
  
  horcm0 11000/udp #Raid Manager instance 0

  See the file /opt/cmd cluster/toolkit/SGCA/Samples/services.example.

- Install MetroCluster with Continuous Access XP on all nodes according to the instructions in the MetroCluster with Continuous Access XP Release Notes.

- Install MC/ServiceGuard if it is not already present, and configure the MC/ServiceGuard cluster according to the procedures outlined in Managing MC/ ServiceGuard. The MAX_CONFIGURED_PACKAGES parameter in the cluster configuration file should be set to 1 or more, depending on the number of packages that will run on the cluster.
Building a Metropolitan Cluster Using MetroCluster/CA
Preparing an MC/ServiceGuard Cluster for MetroCluster /CA

NOTE
Do not create an MC/ServiceGuard package control script template with the `cmmakepkg -s` command or by using SAM. The package control script template `/opt/cmcluster/toolkit/SGCA/sgca.cntl` included with MetroCluster should be used instead.

Creating the Raid Manager Configuration

The Raid Manager configuration file must be edited and customized on each node that is attached to one of the XP Series disk arrays. The file is named using the following convention:

`horcm<instance number>.conf`

All MetroCluster packages must use the same Raid Manager instance, and must be configured in the same configuration file. In the examples in this chapter, instance zero is assumed, which is configured in file `horcm0.conf`.

Here are the steps to follow for creating the configuration:

1. Copy the default Raid Manager configuration file to an instance-specific name:

   ```
   # cp /etc/horcm.conf /etc/horcm0.conf
   ```

2. Create a minimum Raid Manager configuration file by editing the following sections of the file created in the previous step:

   ```
   HORCM_MON  Enter the host-name of the system on which you are editing and the TCP/IP port number specified for this Raid Manager instance in the /etc/services file.
   
   HORCM_CMD  Enter the primary and alternate link device file names for both primary and redundant command devices (for a total of four raw device file names).
   ```

WARNING
Make sure that the redundant command device is not on the same physical device as the primary command device. Also, make sure that the two command devices are on different buses inside the XP Series disk array.
3. Start the Raid Manager instance by using the command
horcmstart.sh <instance-#> as in the following example:

   # horcmstart.sh 0

4. Export the environment variable that specifies the Raid Manager
   instance to be used by the Raid Manager commands. For example,
   with the POSIX shell, type:

   # export HORCMINST=0

   Now, you can use Raid Manager commands to get further information
   from the disk arrays.

5. Verify the software revision of the Raid Manager and the firmware
   revision of the XP Series disk array, use the command raidqry -l.
   The Raid Manager software must be at least revision 01.02.03 and
   the firmware must be at least revision C.

6. Obtain a list of the available devices on the disk arrays using the
   raidscan command. This command must be invoked separately for
   each host interface connection to the disk array. For example, if there
   are two Fibre Channel host adapters, you might use the following
   commands:

   # raidscan -p CL1-A
   # raidscan -p CL1-B

   **NOTE**

   There must also be alternate links for each device, and these alternate
   links must be on different buses inside the XP Series disk array. These
   alternate links, for example, may be CL2-E and CL2-F.

   Unless the devices have been previously paired either on this or
   another host, the devices will show up as SMPL (simplex). Paired
   devices will show up as PVOL (primary volume) or SVOL (secondary
   volume).

7. Determine which devices will be used by the application package.
   Define a device group that contains all of these devices. The device
   group name (dev_group) is user-defined and must be the same on
   each host in the MetroCluster that accesses the XP Series disk array.
   It is recommended that you use a name that is easily associated with
   the package. For example, a device group name of "db-payroll" is
   easily associated with the database for the payroll application. A
device group name of “group1” would be more difficult to easily relate to an application. The device group name MUST be unique within the cluster.

The device name (dev_name) is also user-defined and must be the same on each host in the MetroCluster that accesses the XP Series disk array. The device name (dev_name) must be unique among all devices in the cluster. However, the TargetID and LU# fields for each device name may be different on different hosts in the cluster, to allow for different hardware I/O paths on different hosts.

8. Edit the following sections of the Raid Manager configuration file that was created in a previous step:

**HORCM_DEV**
Include the devices and device group used by the application package. Only one device group may be specified for all of the devices that belong to a single application package.

**HORCM_INST**
Supply the names of only those hosts that are attached to the XP Series disk array that is remote from the disk array directly attached to this host. For example, with a MetroCluster of 6 nodes, 2 of which are Arbitrators, you would specify only hosts 3 and 4 in the **HORCM_INST** section. Host 1 would have previously been specified in the **HORCM_MON** section.

See the file horcm0.conf.<sys-name> in /opt/cmcluster/toolkit/SGCA/Samples/ for an example.

9. Restart the Raid Manager instance so that the new information in the configuration file is read. Use the following commands:

```
# horcmshutdown.sh <instance-#>
# horcmstart.sh <instance-#>
```

10. Repeat these steps on each host that will run this particular application package. If a host may run more than one application package, you must incorporate device group and host information for each of these packages. Note that the Raid Manager configuration file must be different for each host, especially for the **HORCM_MON** and **HORCM_INST** fields.

11. The **HORCM_MON** section of the file is unique for each node in all
Preparing an MC/ServiceGuard Cluster for MetroCluster/CA

clusters that are attached to an XP Series disk array. Enter the host name or IP address followed by the name of the Raid Manager instance that is monitoring the MetroCluster packages on that node (horcm0 in the current example).

12. If you have not already done so, use the paircreate command to create the device groups that are listed in the Raid Manager configuration files. See the Raid Manager User's Guide or view the man page for paircreate for more information. Example:

```
# paircreate -g db_payroll -f data -vl -c15
```

**WARNING**

Paired devices must be of compatible sizes and types.

Sample Raid Manager Configuration File

The following is an example of a Raid Manager configuration file for one node (ftsys1).

```
#
# horcm0.conf.ftsys1
# - This is an example Raid Manager configuration file for node ftsys1.
#  Note that this configuration file is for Raid Manager instance 0,
#  which can be determined by the "0" in the filename "horcm0.conf".
#

# Whenever this configuration file is changed, you must stop and restart the # instance of Raid Manager before the changes will be recognized. This can # be done using the following commands:
#  horcmshutdown.sh <instance>
#  horcmstart.sh <instance>
#
# After restarting the Raid Manager instance, you should confirm that there # are no configuration errors reported by running the pairdisplay command # with the "-c" option.
#
# NOTE: The Raid Manager command device (RORCM_CMD) cannot be used for # data storage (it is reserved for private Raid Manager usage).

#************************ HORCM_MON ************************************
#
# The HORCM_MON parameter is used for monitoring and control of device groups # by the Raid Manager.  
# It is used to define the IP address, port number, and paired volume error # monitoring interval for the local host.  
#<ip_address>
# Defines a network address used by the local host. This can be a host name # or an IP address.
```
Building a Metropolitan Cluster Using MetroCluster/CA
Preparing an MC/ServiceGuard Cluster for MetroCluster /CA

# <service>
# Specifies the port name assigned to the Raid Manager communication path,
# which is must also be defined in /etc/services. If a port number, rather
# than a port name is specified, the port number will be used.
# <poll_interval>
# Specifies the interval used for monitoring the paired volumes. By
# increasing this interval, the Raid Manager daemon load is reduced.
# If this interval is set to -1, the paired volumes are not monitored.
# <timeout>
# Specifies the time-out period for communication with the Raid Manager
# server.

HORCM_MON
#ip_address  service  poll_interval(10ms)  timeout(10ms)
ftsys1        horcm0    1000                3000

/********************************* HORCM_CMD **********************************/
# The HORCM_CMD parameter is used to define the special files (raw device
# file names) of the Raid Manager command devices used for the monitoring
# and control of Raid Manager device groups.
# Define the special device files corresponding to two or more command devices
# in order to use the Raid Manager alternate command device feature. An
# alternate command device must be configured, otherwise a failure of a
# single command device could prevent access to the device group.
# Each command device must have alternate links (PVLinks). The first command
# device is the primary command device. The second command device is a
# redundant command device and is used only upon failure of the primary
# command device. The command devices must be mapped to the various host
# interfaces by using the SVP (disk array console) or a remote console.

HORCM_CMD
#Primary            Primary Alt-Link    Secondary           Secondary Alt-link
#dev_name           dev_name            dev_name            dev_name
/dev/rdsk/c4t1d0    /dev/rdsk/c5t1d0    /dev/rdsk/c4t0d1    /dev/rdsk/c5t0d1

/********************************* HORCM_DEV **********************************/
# The HORCM_DEV parameter is used to define the addresses of the physical
# volumes corresponding to the paired logical volume names. Each group
# name is a unique name used by the hosts which will access the volumes.
# The group and paired logical volume names defined here must be the same for
# all other (remote) hosts that will access this device group.
# The hardware SCSI bus, SCSI-ID, and LUNs for the device groups do not need
# to be the same on remote hosts.

HORCM_DEV
#<dev_group>
# This parameter is used to define the device group name for paired logical
# volumes. The device group name is used by all Raid Manager commands for
# accessing these paired logical volumes.
# <dev_name>
# This parameter is used to define the names of the paired logical volumes
# in the device group.
# <port#>
# This parameter is used to define the XP256 port number used to access the
Building a Metropolitan Cluster Using MetroCluster/CA
Preparing an MC/ServiceGuard Cluster for MetroCluster /CA

# physical volumes in the XP256 connected to the "dev_name". Consult your
# XP256 for valid Port numbers to specify here.
# <TargetID>
# This parameter is used to define the SCSI target ID of the physical
# volume on the port specified in "port#".
# <LUN#>
# This parameter is used to define the SCSI logical unit number (LUN) of
# the physical volume specified in "targetID".

HORCM_DEV
#dev_group dev_name port# TargetID LUN#
pkgA pkgA_index CLI-E 0 1
pkgA pkgA_tables CLI-E 0 2
pkgA pkgA_logs CLI-E 0 3
pkgB pkgB_d1 CLI-E 0 4
pkgC pkgC_d1 CLI-E 0 5
pkgD pkgD_d1 CLI-E 0 2

#/************************* HORCM_INST ************************************/
# This parameter is used to define the network address (IP address or host
# name) of the remote hosts which can provide the remote Raid Manager access
# for each of the device group secondary volumes.
# The remote Raid Manager instances are required to get status or provide
# control of the remote devices in the device group. All remote hosts
# must be defined here, so that the failure of one remote host will prevent
# obtaining status.
# 
# <dev_group>
# This is the same device group names as defined in dev_group of HORC_DEV.
# <ip_address>
# This parameter is used to define the network address of the remote hosts
# with Raid Manager access to the device group. This can be either an
# IP address or a host name.
# <service>
# This parameter is used to specify the port name assigned to the Raid
# Manager instance, which must be registered in /etc/services. If this is
# a port number rather than a port name, then the port number will be used.

HORCM_INST
#dev_group ip_address service
pkgA ftsys1a horcm0
pkgA ftsys2a horcm0
pkgB ftsys1a horcm0
pkgB ftsys2a horcm0
pkgC ftsys1a horcm0
pkgC ftsys2a horcm0
pkgD ftsys1a horcm0
pkgD ftsys2a horcm0

Configuring Automatic Raid Manager Startup
After editing the Raid Manager configuration files and installing them
on the nodes that are attached to the XP Series disk arrays, you should configure automatic Raid Manager startup on the same nodes. You do this by editing the rc script `/etc/rc.config.d/raidmgr`. Set the `START_RAIDMGR` parameter to 1, and define `RAIDMGR_INSTANCE` as the number of the Raid Manager instance you are using with MetroCluster. By default, this is zero (0).

An example of the edited startup file is shown below:

```
#*************************** RAIDMANAGER *************************
# MetroCluster with Continuous Access Toolkit script for configuring the
# startup parameters for a HP SureStore E Disk Array XP256 Raid Manager
# instance. The Raid Manager instance must be running before any
# MetroCluster package can start up successfully.
#
# @(#) $Revision: 1.8 $
#
# START_RAIDMGR:  If set to 1, this host will attempt to start up
#                  an instance of the Disk Array XP256 Raid Manager,
#                  which must be running before a MetroCluster package
#                  can be successfully started. If set to 0, this host
#                  will not attempt to start the Raid Manager.
#                  
# RAIDMGR_INSTANCE  This is the instance number of the Raid Manager
#                  instance to be started by this script. The instance
#                  number specified here must be the same as the
#                  instance number specified in the MetroCluster
#                  package control script.
#                  Consult your Raid Manager documentation for more
#                  information on Raid Manager instances.
#                  
# See the MetroCluster and Raid Manager documentation for more information
# on configuring this script.
#
START_RAIDMGR=0
RAIDMGR_INSTANCE=0
```

**Verifying the XP Series Disk Array Configuration**

Use the following checklist to verify the configuration.
Figure 3-9 XP Series Disk Array Configuration Checklist

☐ Raid Manager XP software installed on all nodes.
☐ PVOLs and SVOLs created for use with all packages.
☐ /etc/horcm0.conf created for all nodes with appropriate differences.
☐ Raid Manager automatic startup configured on all nodes.
☐ Raid Manager running on all nodes.
☐ Primary and alternate command devices configured on all nodes.

Creating and Exporting Volume Groups

Use the following procedure to create volume groups and export them for access by other nodes. The sample script mk1VGs in the Samples directory can be modified to automate these steps.

1. Define the appropriate Volume Groups on each node that might run the application package. Use the commands:

   # mkdir /dev/vg xx
   # mknod /dev/vg xx/group c 64 0x nn 0000

   where the name /dev/vg xx and the number nn are unique within the cluster.

2. Create volume groups only on the primary system. Use the vgcreate and the vgextend command, specifying the appropriate HP-UX device file names.

3. Use the vgexport command with the -p option to export the VGs on the primary system without removing the HP-UX device files:

   # vgchange -a n vname
   # vgexport -v -s -p -m mapfilename vname

   Make sure that you copy the map files to all of the nodes. The sample script Samples/ftpit shows a semi-automated way (using ftp) to copy the files. You need only enter the password interactively.
Preparing an MC/ServiceGuard Cluster for MetroCluster/CA

Importing Volume Groups on Other Nodes

Use the following procedure to import volume groups. The sample script `mk2imports` can be modified to automate these steps.

1. Import the VGs on all of the other systems that will run the MC/ServiceGuard package, and back up the configuration. Use the following commands:
   ```
   # vgimport -v -s -m mapfilename vgname
   ```
2. Back up the configuration. Use the following commands:
   ```
   # vgchange -a y vgname
   # vgcfgbackup vgname
   # vgchange -a n vgname
   ```
   See the sample script `Samples/mk2imports`.

---

**NOTE**

Exclusive activation must be used for all volume groups associated with packages that use the XP Series disk array. The design of MetroCluster/CA assumes that only one system in the cluster will have a VG activated at a time.

---

Configuring PV Links

The examples in the previous sections show the use of the `vgimport` and `vgexport` commands with the `-s` option. Also, the `mk1VGs` script uses a `-s` in the `vgexport` command, and the `mk2imports` script uses a `-s` in the `vgimport` command. You may wish to remove this option from both commands if you are using PV links. The `-s` option to the `vgexport` command saves the volume group id (VGID) in the map file, but it does not preserve the order of PV links. To specify the exact order of PV links, do not use the `-s` option with `vgexport`, and in the `vgimport` command, enter the individual links in the desired order, as in the following example:

```
# vgimport -v -m mapfilename vgname linkname1 linkname2
```
Configuring Packages for Automatic Disaster Recovery

When you have completed the following steps, packages will be able to automatically fail over to an alternate node in another data center and still have access to the data that they need in order to operate.

This procedure must be repeated on all the cluster nodes for each MC/ServiceGuard package so the application can fail over to any of the nodes in the cluster. Customizations include setting environment variables and supplying customer-defined run and halt commands, as appropriate. The package control script must also be customized for the particular application software that it will control. Consult Managing MC/ServiceGuard for more detailed instructions on how to start, halt, and move packages and their services between nodes in a cluster. For ease of troubleshooting, you can configure and test one package at a time.

1. Create a directory `/etc/cmcluster/pkgsname` for each package:
   
   ```
   # mkdir /etc/cmcluster/pkgsname
   
   Copy `/opt/cmcluster/toolkit/SGCA/sgcapkg.cntl` to the package directory, naming it `pkgsname.cntl`:
   
   # cp /opt/cmcluster/toolkit/SGCA/sgcapkg.cntl \
   /etc/cmcluster/pkgsname/pkgsname.cntl
   ```

2. Create a package configuration file with the commands:
   
   ```
   # cd /etc/cmcluster/pkgsname
   
   # cmmakepkg -p pkgsname.config
   ```

   Customize the package configuration file as appropriate to your application. Be sure to include the pathname of the control script (`/etc/cmcluster/pkgsname/pkgsname.cntl`) for the `RUN_SCRIPT` and `HALT_SCRIPT` parameters.

**NOTE**

Do not create an MC/ServiceGuard package control script template with the `cmmakepkg -s` command or by using SAM. The package control script template included with this product should be used instead.
Building a Metropolitan Cluster Using MetroCluster/CA

Configuring Packages for Automatic Disaster Recovery

3. In the <pkgname>.config file, list the node names in the order in which you want the package to fail over. It is recommended for performance reasons, that you have the package fail over locally first, then to the remote data center.

   Set the value of RUN_SCRIPT_TIMEOUT in the package configuration file to NO_TIMEOUT or to a large enough value to take into consideration the extra startup time required to obtain status from the XP Series disk array.

   If you are using a fence level of ASYNC, then the RUN_SCRIPT_TIMEOUT should be greater than the value of HORCTIMEOUT in the package control script (see step 4g below).

   NOTE
   If you are using the EMS disk monitor as a package resource, you must not use NO_TIMEOUT. Otherwise, package shutdown will hang if there is not access from the host to the package disks.

   This toolkit may increase package startup time by 5 minutes or more. Packages with many disk devices will take longer to start up than those with fewer devices due to the time needed to get device status from the XP Series disk array. Clusters with multiple packages that use devices on the XP Series disk array will all cause package startup time to increase when more than one package is starting at the same time.

4. Edit the package control script <pkgname>.cntl as follows:

   a. If necessary, add the path where the RaidManager software binaries have been installed to the PATH environment variable. If the software is in the usual location, /usr/bin, you can just uncomment the line in the script.

   b. Uncomment the behavioral configuration environment variables starting with AUTO_. It is recommended that you retain the default values of these variables unless you have a specific business requirement to change them. See Appendix A for an explanation of these variables.

   c. Uncomment the PKGDIR variable and set it to the full path name of the directory where the control script has been placed. This directory, which is used for status data files, must be unique for each package. For example, set PKGDIR to
Building a Metropolitan Cluster Using MetroCluster/CA
Configuring Packages for Automatic Disaster Recovery

/etc/cmcluster/package_name, removing any quotes around the file names.

d. Uncomment the DEVICE_GROUP variable and set it to this package's Raid Manager device group name, as specified in the Raid Manager configuration file.

e. Uncomment the HORCMINST variable and set it to the Raid Manager instance name used by MetroCluster/CA.

f. Uncomment the FENCE variable and set it to either ASYNC, NEVER, or DATA according to your business requirements or special MetroCluster requirements. This variable is used for two purposes:
   • To compare with the actual fence level returned by the array.
   • To set the fence level when paircreate operations are performed.

g. If you are using asynchronous data replication, set the HORCTIMEOUT variable to a value greater than the side file timeout value configured with the Service Processor (SVP), but less than the RUN_SCRIPT_TIMEOUT set in the package configuration file. The default setting is the side file timeout value + 60 seconds.

h. Uncomment the DC1HOST array variable and set the elements to the node names of the systems on the local side of the ESCON link. The order of the node names is not important.

i. Uncomment the DC2HOST array variable and set the elements to the node names of the systems on the remote side of the ESCON link. The order of the node names is not important.

j. Uncomment the CLUSTER_TYPE variable and set it to METRO. (The value CONTINENTAL is for use with ContinentalClusters, described in Chapter 5.)

5. Customize the control script as appropriate to your application using the guidelines in Managing MC/ServiceGuard. Standard MC/ServiceGuard package customizations include modifying the VG, LV, FS, IP, SUBNET, SERVICE_NAME, SERVICE_CMD and SERVICE_RESTART parameters. Be sure to set LV_UMOUNT_COUNT to 1 or greater.

6. Add customer-defined run and halt commands in the appropriate places according to the needs of the application. See Managing
Building a Metropolitan Cluster Using MetroCluster/CA

Configuring Packages for Automatic Disaster Recovery

MC/ServiceGuard for more information on these functions.

7. After customizing the control script file and before starting up the package, do a syntax check on the control script using the following command (be sure to include the -n option to perform syntax checking only):

```
# sh -n <pkgname.cntl>
```

If any messages are returned, you should correct the syntax errors.

8. Check the configuration using the `cmcheckconf -P pkgname.config`, then apply the MC/ServiceGuard configuration using the `cmapplyconf -P pkgname.config` command or SAM.

9. Distribute MetroCluster/CA control script files to other nodes in the cluster by using `ftp` or `rcp`:

```
# rcp -p /etc/cmcluster/pkgname/pkgname.cntl \
other_node:/etc/cmcluster/pkgname/pkgname.cntl
```

See the example script `Samples/ftpit` to see how to semi-automate the copy using `ftp`. This script assumes the package directories already exist on all nodes.

Using `ftp` may be preferable at your organization, since it does not require the use of a `.rhosts` file for root. Root access via `.rhosts` may create a security issue.

10. Verify that each node in the MC/ServiceGuard cluster has the following files in the directory `/etc/cmcluster/pkgname`:

```
pkgname.cntl   MetroCluster/CA package control script
pkgname.config ServiceGuard package ASCII configuration file
pkgname.sh    Package monitor shell script, if applicable
other files   Any other scripts you use to manage MC/ServiceGuard packages
```

The MC/ServiceGuard cluster is ready to automatically switch packages to nodes in remote data centers using MetroCluster/CA.
Maintaining a Cluster that uses MetroCluster/CA

While the cluster is running, manual changes of state for devices on the XP Series disk array can cause the package to halt due to unexpected conditions or can cause the package to not start up after a failover. In general, it is recommended that no manual changes of state be performed while the package and the cluster are running.

NOTE

Manual changes can be made when they are required to bring the device group into a “protected” state. For example, if a package starts up with volumes in a Simplex state due to a disaster, you can run the paircreate command to re-establish disaster protection when the other disk array is available again. This can be done while the package is running.

Viewing the Progress of Copy Operations

While a copy is in progress between XP systems (that is, the volumes are in a COPY state), you can see the progress of the copy by viewing the % column in the output of the pairdisplay command:

```
# pairdisplay -g pkgB -fc -CLI
```

This display shows that 79% of a current copy operation has completed. Synchronous fence levels (NEVER and DATA) show 100% in this column when the volumes are in a PAIR state.

Viewing Side File Size

If you are using asynchronous data replication, you can see the current size of the side file when the volumes are in a PAIR state by using the pairdisplay command. The following output, obtained during normal cluster operation, shows the percentage of the side file that is full:
Building a Metropolitan Cluster Using MetroCluster/CA

Maintaining a Cluster that uses MetroCluster/CA

```bash
# pairdisplay -g pkgB -fc -CLI
Group   PairVol L/R   Port# TID LU  Seq# LDEV# P/S Status Fence    %  P-LDEV# M
pkgB    pkgD-disk0 L  CL1-C  0   3 35422  463  P-VOL PAIR ASYNC 35 3 -
pkgB    pkgD-disk0 R  CL1-F  0   3 35663    3  S-VOL PAIR ASYNC 0 463 -
```

This output shows that 35% of the side file is full.

When volumes are in a COPY state, the % column shows the progress of the copying between the XP frames, until it reaches 100%, at which point the display reverts to showing the side file usage in the PAIR state.

**Normal Maintenance**

There might be situations when the package has to be taken down for maintenance purposes without having the package move to another node. The following procedure is recommended for normal maintenance of the MetroCluster/CA:

1. Stop the package with the appropriate MC/ServiceGuard command.
   ```bash
   # cmhaltpkg pkgname
   ```
2. Distribute the MetroCluster with Continuous Access XP configuration changes.
   ```bash
   # cmapplyconf -P pkgname.config
   ```
3. Start the package with the appropriate MC/ServiceGuard command:
   ```bash
   # cmmodpkg -e pkgname
   ```

Planned maintenance is treated the same as a failure by the cluster. If you take a node down for maintenance, package failover and quorum calculation is based on the remaining nodes. Make sure that nodes are taken down evenly at each site, and that enough nodes remain on-line to form a quorum if a failure occurs. See “Example Failover Scenarios with Two Arbitrators” on page 59.

**Resynchronizing**

After certain failures, data is no longer remotely protected. In order to restore disaster-tolerant data protection after repairing or recovering from the failure, you must manually run the command `pairresync`. This command must successfully complete for disaster-tolerant data...
Building a Metropolitan Cluster Using MetroCluster/CA

Maintaining a Cluster that uses MetroCluster/CA

Following is a partial list of failures that require running `pairresync` to restore disaster-tolerant data protection:

- Failure of all CA (ESCON) links without restart of the application
- Failure of all CA links with Fence Level “DATA” with restart of the application on a primary host
- Failure of the entire secondary Data Center for a given application package
- Failure of the secondary XP Series disk array for a given application package while the application is running on a primary host

Following is a partial list of failures that require full resynchronization to restore disaster-tolerant data protection. Full resynchronization is automatically initiated for these failures by moving the application package back to its primary host after repairing the failure:

- Failure of the entire primary data center for a given application package
- Failure of all of the primary hosts for a given application package
- Failure of the primary XP Series disk array for a given application package
- Failure of all CA links with restart of the application on a secondary host

Pairs must be manually recreated if both the primary and secondary XP Series disk array are in SMPL (simplex) state. Make sure you periodically review the files `syslog.log` and `/etc/cmcluster/pkgname/pkgname.log` for messages, warnings and recommended actions. You should particularly review these files after system, data center and/or application failures.

Full resynchronization must be manually initiated after repairing the following failures:

- Failure of the secondary XP Series disk array for a given application package followed by application startup on a primary host
- Failure of all CA links with Fence Level NEVER and ASYNC with restart of the application on a primary host
Using the `pairresync` Command

The `pairresync` command can be used with special options after a failover in which the recovery site has started the application and has processed transaction data on the disk at the recovery site, but the disks on the primary site are intact. After the ESCON link is fixed, you use the `pairresync` command in one of the following two ways depending on which site you are on:

- `pairresync -swapp` — from the primary site.
- `pairresync -swaps` — from the failover site.

These options take advantage of the fact that the recovery site maintains a bit-map of the modified data sectors on the recovery array. Either version of the command will swap the personalities of the volumes, with the PVOL becoming the SVOL and SVOL becoming the PVOL. With the personalities swapped, any data that has been written to the volume on the failover site (now PVOL) are then copied back to the SVOL now running on the primary site. During this time the package continues running on the failover site. After resynchronization is complete, you can halt the package on the failover site, and restart it on the primary site. MetroCluster will then swap the personalities between the PVOL and the SVOL, returning PVOL status to the primary site.

---

**NOTE**

The preceding steps are automated provided the default value of 1 is being used for the auto variable `AUTO_PSUEPSUS`. Once the ESCON link failure has been fixed, the user only needs to halt the package on the recovery cluster and restart on the primary cluster. However, if you want to reduce the amount of application downtime, you should manually invoke `pairresync` before failback.

---

Failback

After resynchronization is complete, you can halt the package on the failover site, and restart it on the primary site. MetroCluster will then swap the personalities between the PVOL and the SVOL, returning PVOL status to the primary site.
4 Building a Metropolitan Cluster Using MetroCluster/SRDF

This chapter describes the high availability toolkit template for MetroCluster with EMC SRDF (hereafter known as MetroCluster/SRDF), which allows you to install and configure an MC/ServiceGuard metropolitan cluster using the EMC Symmetrix disk array. It is assumed that readers are already familiar with MC/ServiceGuard configuration tasks, MC/ServiceGuard metropolitan cluster concepts and installation procedures, EMC Symmetrix disk arrays and SRDF concepts, and Symmetrix Command Line Interface configuration and use.

The topics discussed in this chapter are:

• Overview of MetroCluster/SRDF
• Designing a Disaster Tolerant Architecture for use with MetroCluster/SRDF
• Preparing a Cluster for MetroCluster/SRDF
• Configuring MC/ServiceGuard Packages for Automatic Disaster Recovery
• Setting up M by N Configurations
• Maintaining a Cluster that Uses MetroCluster/SRDF
Overview of MetroCluster/SRDF

MetroCluster is a set of scripts that work in an MC/ServiceGuard cluster to automate failover to alternate nodes in the case of disaster in a campus or metropolitan cluster. The MetroCluster/SRDF template contains the following files, installed in /opt/cmcluster/toolkit/SGSRDF:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>srdfpkg.cntl</td>
<td>The MetroCluster/SRDF control script template. This template must be customized for the specific EMC Symmetrix and HP 9000 host system configuration. Copies of this template must be customized for each separate MC/ServiceGuard package.</td>
</tr>
<tr>
<td>Samples</td>
<td>A directory containing sample convenience shell scripts that must be edited before using. These shell scripts may help to automate some configuration tasks. These scripts are contributed, and not supported. This directory also contains sample Raid Manager configuration files and other examples.</td>
</tr>
</tbody>
</table>

MetroCluster/SRDF has to be installed on all nodes that will run a ServiceGuard package that accesses data on an EMC Symmetrix where the data are replicated to a second Symmetrix using the SRDF facility. In the event of node failure, the integration of MetroCluster/SRDF with the package will allow the application to fail over in the following ways:

- among local host systems that are attached to the same EMC Symmetrix.
- between one system that is attached locally to its EMC Symmetrix and another “remote” host that is attached locally to the other EMC Symmetrix.

MetroCluster with EMC SRDF is specifically for configuring one or more MC/ServiceGuard packages whose data reside on EMC Symmetrix ICDAs (Integrated Cache Disk Arrays) replicated with SRDF (Symmetrix Remote Data Facility).

MetroCluster with EMC SRDF can be used in a campus or metropolitan cluster configuration with nodes in a loop of up to 100 km. Symmetrix configurations can be either 1 by 1 (one Symmetrix at each data center)
Building a Metropolitan Cluster Using MetroCluster/SRDF

Overview of MetroCluster/SRDF

or M by N (one or two Symmetrix frames at each data center).

Configuration of MetroCluster/SRDF must be done on all the cluster
nodes, as is done for any other MC/ServiceGuard package. To use
MetroCluster/SRDF, Symmetrix host-based software for control and
status of the EMC Symmetrix disk arrays must also be installed and
configured on each HP 9000 host system that might execute the
application package.
Designing a Disaster Tolerant Architecture for use with MetroCluster/SRDF

MetroCluster with EMC SRDF is designed for use in a campus cluster or metropolitan cluster environment within the 100 km loop limit of the FDDI network.

All nodes must be members of a single MC/ServiceGuard cluster. Two configurations are supported:

- A single data center without arbitrators (not disaster tolerant.)
- A Three-Data-Center architecture with one or two Arbiter systems. See Figure 4-1 on page 91.

Following are the disaster tolerant architecture requirements:

- In the disaster tolerant cluster architecture, it is expected that each data center is self-contained such that the loss of one data center does not cause the entire cluster to fail. It is important that all Single Points of Failure (SPOF) be eliminated so that surviving systems continue to run in the event that one or more systems fail.
- It is also expected that the networks between the data centers are redundant and routed in such a way that the loss of any one data center does not cause the network between surviving data centers to fail.
- Exclusive Volume Group activation must be used for all VGs associated with packages that use the Symmetrix. The design of the MetroCluster with EMC SRDF script assumes that only one system in the cluster will have a VG activated at any time.

Single Data Center

A single data center architecture is supported, but it is not a true disaster tolerant architecture. If the entire data center fails, there will be no automated failover. This architecture is only valid for protecting data through data replication, and for protecting against multiple node failures.
Building a Metropolitan Cluster Using MetroCluster/SRDF

Designing a Disaster Tolerant Architecture for use with MetroCluster/SRDF

Three Data Centers with Arbitrator(s)

This is the recommended and supported disaster tolerant architecture for use with MetroCluster with EMC SRDF. The three-data-center architecture consists of two data centers with an equal number of nodes and a third data center with one or more arbitrator nodes; see Figure 4-1.

Figure 4-1  Three Data Centers with Arbitrators

The local EMC Symmetrix disk array is called the R1 disk for all nodes and packages in a given data center. The remote EMC Symmetrix disk array, where the data is replicated is called the R2 disk. In Figure 4-1 the EMC Symmetrix disk array in data center A is the R1 disk for packages A and B, and the R2 disk for packages C and D.
A and B, and the R2 disk for packages C and D in data center B. Likewise the EMC Symmetrix disk array in data center B is the R1 for packages C and D, and the R2 for packages A and B.

Arbitrators provide similar functionality to the cluster lock disk, and act as tie-breakers for a cluster quorum in case all of the nodes in one data center go down at the same time. Cluster lock devices are not used in the three-data-center architecture because cluster locks cannot be maintained across the SRDF link.

Arbitrators are fully-functioning systems that are members of the cluster and are not usually physically connected to the Symmetrix units. Table 4-1 lists the allowable number of nodes at each data center in a three-data-center configuration, assuming a 16-node maximum cluster size. Figure 4-1 on page 91 shows a three-data-center configuration with two nodes at each site, and two arbitrator nodes.

### Table 4-2 Possible Number of Nodes in a Three Data Center Configuration

<table>
<thead>
<tr>
<th>Primary Data Center A (with Symmetrix)</th>
<th>Primary Data Center B (with Symmetrix)</th>
<th>Arbitrator Data Center C (No Symmetrix)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2*</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>2*</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>2*</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>2*</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>2*</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>2*</td>
</tr>
</tbody>
</table>
Building a Metropolitan Cluster Using MetroCluster/SRDF

Designing a Disaster Tolerant Architecture for use with MetroCluster/SRDF

* Configurations with two arbitrators are preferred because they provide a greater degree of availability, especially in cases when a node is down due to a failure or planned maintenance.

NOTE

In the campus or metropolitan environment, the same number of systems must be present in each of the two data centers (Data Center A and Data Center B) whose systems are connected to the Symmetrix units. There must be either one or two arbitrators in Data Center C.

Arbitrator Node Configuration Rules

Although you can use one arbitrator, having two arbitrators provides greater flexibility in taking systems down for planned outages as well as providing better protection against multiple points of failure:

- Provides local failover capability to applications running on the arbitrator.
- Protects against more MPOF (Multiple Points of Failure).
- Provides for planned downtime of a single system anywhere in the cluster.

If you use a single arbitrator system, special procedures must be followed during planned downtime to remain protected. Systems must be taken down in pairs, one from each of the data centers, so that the MC/ServiceGuard quorum is maintained after a node failure. If the arbitrator itself must be taken down, disaster recovery capability is at risk if one of the other systems fails.

Arbitrator systems can be used to perform important and useful work such as:

- Running mission-critical applications not protected by disaster recovery
- IT/Operations or NetworkNodeManager
- Backup
- Application servers

Calculating a Cluster Quorum

When a cluster initially forms, all systems must be available to form the
Building a Metropolitan Cluster Using MetroCluster/SRDF

Designing a Disaster Tolerant Architecture for use with MetroCluster/SRDF

center (100% Quorum requirement).

A quorum is dynamic and is recomputed after each system failure. For instance, if you start out with an 8-node cluster and two systems fail, that leaves 6 out 8 surviving nodes, or a 75% quorum. The cluster size is reset to 6 nodes. If two more nodes fail, leaving 4 out of 6, quorum is 67%.

Each time a cluster forms, there must be more than 50% quorum to reform the cluster. Cluster lock disks are normally used as the tie-breaker when quorum is exactly 50%. However, cluster lock disks are not supported with MetroCluster with EMC SRDF. Therefore, a quorum of 50% or less will cause the remaining nodes to halt.

Example Failover Scenarios with One Arbiter

Taking a node off-line for planned maintenance is treated the same as a node failure in these scenarios. Study these scenarios to make sure you do not put your cluster at risk during planned maintenance.

Figure 4-2 Failover Scenario with a Single Arbiter

The scenarios in Table 4-3 are based on Figure 4-2 and illustrate possible results if one or more nodes fail in a configuration with a single
Designing a Disaster Tolerant Architecture for use with MetroCluster/SRDF

* Cluster can be manually started with the remaining node.

Table 4-3

<table>
<thead>
<tr>
<th>Node Failure</th>
<th>Quorum</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>arbitrator 1</td>
<td>4 of 5 (80%)</td>
<td>nothing</td>
</tr>
<tr>
<td>node 1</td>
<td>4 of 5 (80%)</td>
<td>pkg A switches</td>
</tr>
<tr>
<td>node 1, then node 2</td>
<td>3 of 4 (75%)</td>
<td>pkg A and B switch</td>
</tr>
<tr>
<td>node 1, 2, then arbitrator 1</td>
<td>2 of 3 (67%)</td>
<td>nothing</td>
</tr>
<tr>
<td>nodes 1, 2, arbitrator 1, then node 3</td>
<td>1 of 2 (50%)</td>
<td>cluster halts*</td>
</tr>
<tr>
<td>arbitrator 1, then node 1</td>
<td>3 of 4 (75%)</td>
<td>pkg A switches</td>
</tr>
</tbody>
</table>

* Cluster can be manually started with the remaining node.

Table 4-4

<table>
<thead>
<tr>
<th>Node Failure</th>
<th>Quorum</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>data center A (nodes 1 and 2)</td>
<td>3 of 5 (60%)</td>
<td>pkg A and B switch to data center B</td>
</tr>
<tr>
<td>data center A, then arbitrator 1</td>
<td>2 of 3 (67%)</td>
<td>pkg A and B switch, then nothing</td>
</tr>
<tr>
<td>data center A and arbitrator 1</td>
<td>2 of 5 (40%)</td>
<td>cluster halts*</td>
</tr>
<tr>
<td>data center A, then arbitrator 1, then node 3</td>
<td>1 of 2 (50%)</td>
<td>cluster halts*</td>
</tr>
<tr>
<td>arbitrator 1, then data center A</td>
<td>2 of 4 (50%)</td>
<td>cluster halts*</td>
</tr>
<tr>
<td>node 3, then data center A</td>
<td>2 of 4 (50%)</td>
<td>cluster halts*</td>
</tr>
<tr>
<td>data center B</td>
<td>3 of 5 (60%)</td>
<td>pkg C and D switch to data center A</td>
</tr>
<tr>
<td>data center C</td>
<td>4 of 5 (80%)</td>
<td>nothing</td>
</tr>
</tbody>
</table>

* Cluster can be manually started with the remaining node.
Building a Metropolitan Cluster Using MetroCluster/SRDF
Designing a Disaster Tolerant Architecture for use with MetroCluster/SRDF

With a single arbitrator node, the cluster is at risk each time a node fails or you take one node down for planned maintenance.

**Example Failover Scenarios with Two Arbitrators**

Having two arbitrator nodes adds extra protection during nodes failures and allows you to do planned maintenance on arbitrator nodes without losing the cluster should a disaster occur.

**Figure 4-3** Failover Scenario with Two Arbitrators

The scenarios in Table 4-5 illustrate possible results if a data center or one or more nodes fail in a configuration with two arbitrators. Note that 3 of the 4 scenarios that caused a cluster halt with a single arbitrator, do not cause a cluster halt with two arbitrators.

**Table 4-5** Data Center Failure Scenarios with Two Arbitrators

<table>
<thead>
<tr>
<th>Node Failure</th>
<th>Quorum</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>data center A (nodes 1 and 2)</td>
<td>4 of 6 (67%)</td>
<td>pkg A and B switch to data center B</td>
</tr>
<tr>
<td>data center A, then arbitrator 1</td>
<td>3 of 4 (75%)</td>
<td>pkg A and B switch, then nothing</td>
</tr>
<tr>
<td>data center A and arbitrator 1</td>
<td>3 of 6 (50%)</td>
<td>cluster halts*</td>
</tr>
<tr>
<td>data center A, then arbitrator 1, then node 3</td>
<td>2 of 3 (67%)</td>
<td>pkg A, B, and C switch</td>
</tr>
<tr>
<td>arbitrator 1, then data center A</td>
<td>3 of 5 (60%)</td>
<td>pkg A and B switch to data center B</td>
</tr>
</tbody>
</table>
Table 4-5 Data Center Failure Scenarios with Two Arbitrators

<table>
<thead>
<tr>
<th>Node Failure</th>
<th>Quorum</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>node 3, then data center A</td>
<td>3 of 5 (60%)</td>
<td>pkg A and B switch to data center B</td>
</tr>
<tr>
<td>data center B</td>
<td>4 of 6 (67%)</td>
<td>pkg C and D switch to data center A</td>
</tr>
<tr>
<td>data center C</td>
<td>4 of 6 (67%)</td>
<td>nothing</td>
</tr>
</tbody>
</table>

* Cluster can be manually started with the remaining node.

**Disaster Tolerant Checklist**

Use this checklist to make sure you have adhered to the disaster tolerant architecture guidelines for a three-data-center configuration.

**Figure 4-4 Disaster Tolerant Checklist**

- Each data center must have the same number of nodes to maintain a proper quorum in case an entire data center fails.
- Arbitrator nodes should be located in a separate data center from either of the primary data centers (A or B).
- The elements in each data center including nodes, disks, network components, and climate control should be on separate power circuits.
- Each node must be configured with PV links.
- Each EMC Symmetrix disk array must be configured with redundant SRDF links.
- At least two networks must function as the cluster heartbeat.
- All redundant cabling for network, heartbeat, and SRDF connections should be routed using different physical paths.
Building a Metropolitan Cluster Using MetroCluster/SRDF

Designing a Disaster Tolerant Architecture for use with MetroCluster/SRDF

Cluster Configuration Worksheet

Use this cluster configuration worksheet either in place of, or in addition to the worksheet provided in the Managing MC/ServiceGuard manual. If you have already completed an MC/ServiceGuard cluster configuration worksheet, you only need to complete the first part of this worksheet.

**Figure 4-5**

<table>
<thead>
<tr>
<th>Name and Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster Name:</td>
</tr>
<tr>
<td>Data Center A Name and Location:</td>
</tr>
<tr>
<td>Node Names:</td>
</tr>
<tr>
<td>Data Center B Name and Location:</td>
</tr>
<tr>
<td>Node Names:</td>
</tr>
<tr>
<td>Arbitrator Data Center C Name and Location:</td>
</tr>
<tr>
<td>Arbitrator Node Names:</td>
</tr>
<tr>
<td>Maximum Configured Packages:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Subnets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heartbeat IP Addresses:</td>
</tr>
<tr>
<td>Non-Heartbeat IP Addresses:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Timing Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heartbeat Interval:</td>
</tr>
<tr>
<td>Node Time-out:</td>
</tr>
<tr>
<td>Network Polling Interval:</td>
</tr>
<tr>
<td>AutoStart Delay:</td>
</tr>
</tbody>
</table>
Package Configuration Worksheet

Use this package configuration worksheet either in place of, or in addition to the worksheet provided in the Managing MC/ServiceGuard manual. If you have already completed an MC/ServiceGuard cluster configuration worksheet, you only need to complete the first part of this worksheet.

Figure 4-6

Package Configuration File Data:

Package Name: ____________________________
Primary Node: ____________________________ Data Center: ____________________________
First Failover Node: _____________________ Data Center: ____________________________
Second Failover Node: __________________ Data Center: ____________________________
Third Failover Node: _____________________ Data Center: ____________________________
Fourth Failover Node: ___________________ Data Center: ____________________________
Package Run Script: ____________________ Timeout: _______________________________
Package Halt Script: ____________________ Timeout: ______________________________
Maximum Configured Packages: ____________________________

Symmetrix Device Groups

R1 Symmetrix Device Group: ____________________________
R2 Symmetrix Device Group: ____________________________
R1Host[0]: ______________ Data Center: ____________________________
R1Host[1]: ______________ Data Center: ____________________________
R1Host[2]: ______________ Data Center: ____________________________
R2Host[0]: ______________ Data Center: ____________________________
R2Host[1]: ______________ Data Center: ____________________________
R2Host[2]: ______________ Data Center: ____________________________

Package Control Script Data

VG[0]: _______LV[0]: _______FS[0]: _______FS_MOUNT_OPT[0]:
VG[1]: _______LV[1]: _______FS[1]: _______FS_MOUNT_OPT[1]:
VG[2]: _______LV[2]: _______FS[2]: _______FS_MOUNT_OPT[2]:
IP[0]: ___________________ SUBNET[0]: __________________
IP[1]: ___________________ SUBNET[1]: __________________
X.25 Resource Name: ____________________________
Service Name: _______________ Run Command: _______________ Retries: __ RetryTime: __
Service Fail Fast Enabled?: _______________ Service Halt Timeout: _______________
Service Name: _______________ Run Command: _______________ Retries: __ RetryTime: __
Service Fail Fast Enabled?: _______________ Service Halt Timeout: _______________
Preparing a Cluster for MetroCluster/SRDF

When the following procedures are completed, an adoptive node will be able to access the data belonging to a package after it fails over. Use the convenience scripts in the /opt/cmcluster/toolkits/SGSRDF/Samples to automate some of the tasks in the following sections:

- `mk3symgrps.nodename` — to create EMC Symmetrix device groups.
- `mk4gatekpr.nodename` — to create gatekeeper devices.
- `mk2imports` — to create volume groups.
- `ftpit` — to copy the configuration to other nodes in the cluster.
- `pre.cmquery` — to split SRDF links before applying the package configuration.
- `post.cmapply` — to restore SRDF links after applying the package configuration.

These scripts should be copied from /opt/cmcluster/toolkits/SGSRDF to another directory, such as /etc/cmcluster/SRDF.

Installing the Necessary Software

Before any configuration can begin, you need to make sure the following software is installed on all nodes:

- Symmetrix SymCLI command line interface that allows you to manage the Symmetrix disks from the node.
- Symmetrix PowerPath software if you are building an M by N configuration.
- If using HP-PB F/W SCSI host adapters in HP-UX 10.20, install and configure the HP-PB F/W SCSI Pass-Through driver. Otherwise, the SymCLI commands will not work properly. In case of improper configuration, you will see error messages in the package log file like: Bad File Number.
- MetroCluster with EMC SRDF should be installed on all nodes according to the instructions in the MetroCluster with EMC SRDF Release Notes.
Setting up Hardware for 1 by 1 Configurations

Ensure that the Symmetrix disk arrays are correctly cabled using PV links to each node in the cluster that will run packages that access data on the Symmetrix.

See the SymCLI manual for instructions on creating the appropriate pseudo device files.

• R1 and R2 devices must have been correctly defined and assigned to the appropriate nodes in the internal configurations that is downloaded by EMC support staff.

Figure 4-7 EMC R1 and R2 Definitions

- R1 devices are locally protected (RAID 1 or RAID S).
- R2 devices are locally protected (RAID 1, RAID S or BCV).
- Only Synchronous Mode is supported; Adaptive Copy must be disabled.
- Domino Mode is recommended to ensure data currency on each EMC Symmetrix disk array.
Preparing a Cluster for MetroCluster/SRDF

- To minimize contention, each package should be assigned its own, unique gatekeeper device. See “Configuring Gatekeeper Devices” on page 114.

**Setting up Hardware for M by N Configurations**

You can configure up to four Symmetrix disk arrays in the following combinations:

- An array in Data Center A connected to one array in Data Center B.
- An array in Data Center A connected to two arrays in Data Center B.
- Two arrays in Data Center A connected to an array in Data Center B.
- Two arrays in Data Center A connected to two arrays in Data Center B.

Figure 4-8 shows a 2 by 1 configuration with BCVs.
BCVs at Data Center B for pkg A and pkg B.

Figure 4-9 shows a 2 by 2 configuration with R1 volumes for pkg A and pkg B on the Symmetrix frames located in Data Center A and R2 volumes and BCVs at Data Center B. Many of the examples given later in this chapter are based on this configuration.

Figure 4-9 2 by 2 Node and Data Center Configuration

Figure 4-10 below shows a bidirectional 2 by 2 configuration with additional packages on node3 and node4, and R1 and R2 volumes at both data centers. In this configuration, R1 volumes for pkg A and pkg B are at Data Center A, and R2 volumes are at Data Center B. R1 volumes for pkg C and pkg D are at Data Center B, and R2 volumes are at Data Center A.
Grouping the Symmetrix Devices at Each Data Center

The use of R1/R2 devices in M by N configurations of multiple Symmetrix frames is enabled by means of consistency groups. A consistency group is a set of Symmetrix RDF devices that are configured to act in unison to maintain the integrity of a database. Because MetroCluster/SRDF works at the device group level, the consistency group is implemented and managed as a single device group even though it spans multiple Symmetrix frames. Consistency groups are managed by the EMC PowerPath product. When PowerPath is installed, the Symmetrix tracks the I/Os that are written to the devices in the consistency group. If an I/O cannot be written to a remote Symmetrix due to a failure of a remote device or an RDF link, the data flow to the other Symmetrix will be halted in less than one second. Once mirroring is resumed, any updates will propagate with normal SRDF operation.

Figure 4-11 shows that when there is a break in the links between two of
Building a Metropolitan Cluster Using MetroCluster/SRDF

Preparing a Cluster for MetroCluster/SRDF

the Symmetrix frames, the use of consistency groups (dashed oval lines) ensures that the other two links are also suspended.

**Figure 4-11** 2 X 2 Node and Data Center Configuration with Consistency Groups

---

**Building the Symmetrix CLI Database**

The Symmetrix CLI (Command Line Interface) should be installed on all nodes running packages that use data on the EMC Symmetrix disk arrays. Create the SymCLI database on each system using the following steps. For complete information, refer to the Symmetrix SymCLI manual.

Issue the following command on each node after the hardware is installed:

```
# symcfg discover
```
Preparing a Cluster for MetroCluster/SRDF

This builds the CLI database on the node.

You can display what is in the SymCLI database with the commands:

- `symdg list`
- `symld -g symdevgrpname list`
- `symgate list`

If you have not configured the SymCLI database, you will see an error:

The Symmetrix configuration could not be loaded for a locally attached Symmetrix

**NOTE**

Make sure that you do not set the `SYMCLI_SID` and `SYMCLI_DG` environment variables before running the `symcfg` command. These environment variables limit the amount of information gathered when the SymCLI database is created, and will not give you a complete database.

Also, the `SYMCLI_OFFLINE` variable should not be set. This environment variable disables the command line interface.

---

**Determining Symmetrix Device Names on Each Node**

To correctly specify the device file names when creating Symmetrix device groups, you need to know how the HP-UX device files map to the R1 and R2 Symmetrix devices. Use the following steps to gather the necessary information.

1. Obtain a list of data for the Symmetrix devices available, using the following command on each node without any options:

   ```
   #syminq
   ```

   Sample output from both the R1 and R2 sides is shown in Figure 4-12 and Figure 4-13.
2. You need the following information from these listings for each Symmetrix logical device:
   - HP-UX device file name (e.g. /dev/rdsk/c3t3d2).
   - device type (R1, R2, BCV, GK, or blank)
   - Symmetrix serial number (e.g. 50006161), useful in matching the

<table>
<thead>
<tr>
<th>Device Name</th>
<th>Type</th>
<th>Vendor ID</th>
<th>Rev</th>
<th>Ser Num</th>
<th>Cap(KB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dev/rdsk/c0t0d0</td>
<td>R1</td>
<td>EMC</td>
<td>5264</td>
<td>95004160</td>
<td>4418880</td>
</tr>
<tr>
<td>/dev/rdsk/c0t0d1</td>
<td>R1</td>
<td>EMC</td>
<td>5264</td>
<td>95005160</td>
<td>4418880</td>
</tr>
<tr>
<td>/dev/rdsk/c0t0d2</td>
<td>R2</td>
<td>EMC</td>
<td>5264</td>
<td>95006160</td>
<td>4418880</td>
</tr>
<tr>
<td>/dev/rdsk/c0t0d3</td>
<td>R2</td>
<td>EMC</td>
<td>5264</td>
<td>95007160</td>
<td>4418880</td>
</tr>
<tr>
<td>/dev/rdsk/c0t1d0</td>
<td>BCV</td>
<td>EMC</td>
<td>5264</td>
<td>95024160</td>
<td>4418880</td>
</tr>
<tr>
<td>/dev/rdsk/c0t1d1</td>
<td>BCV</td>
<td>EMC</td>
<td>5264</td>
<td>95025160</td>
<td>4418880</td>
</tr>
<tr>
<td>/dev/rdsk/c0t1d0</td>
<td>GK</td>
<td>EMC</td>
<td>5264</td>
<td>95040160</td>
<td>2880</td>
</tr>
<tr>
<td>/dev/rdsk/c0t1d1</td>
<td>GK</td>
<td>EMC</td>
<td>5264</td>
<td>95041160</td>
<td>2880</td>
</tr>
<tr>
<td>/dev/rdsk/c0t1d2</td>
<td>GK</td>
<td>EMC</td>
<td>5264</td>
<td>95042160</td>
<td>2880</td>
</tr>
<tr>
<td>/dev/rdsk/c1t2d0</td>
<td>R1</td>
<td>EMC</td>
<td>5264</td>
<td>95004320</td>
<td>4418880</td>
</tr>
<tr>
<td>/dev/rdsk/c1t2d1</td>
<td>R1</td>
<td>EMC</td>
<td>5264</td>
<td>95005320</td>
<td>4418880</td>
</tr>
<tr>
<td>/dev/rdsk/c1t2d2</td>
<td>R2</td>
<td>EMC</td>
<td>5264</td>
<td>95006320</td>
<td>4418880</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Device Name</th>
<th>Type</th>
<th>Vendor ID</th>
<th>Rev</th>
<th>Ser Num</th>
<th>Cap(KB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dev/rdsk/c4t0d0</td>
<td>R2</td>
<td>EMC</td>
<td>5264</td>
<td>50014321</td>
<td>4418880</td>
</tr>
<tr>
<td>/dev/rdsk/c4t0d1</td>
<td>R2</td>
<td>EMC</td>
<td>5264</td>
<td>50015321</td>
<td>4418880</td>
</tr>
<tr>
<td>/dev/rdsk/c4t0d2</td>
<td>R1</td>
<td>EMC</td>
<td>5264</td>
<td>50016321</td>
<td>4418880</td>
</tr>
<tr>
<td>/dev/rdsk/c4t0d3</td>
<td>R1</td>
<td>EMC</td>
<td>5264</td>
<td>50017321</td>
<td>4418880</td>
</tr>
<tr>
<td>/dev/rdsk/c4t1d0</td>
<td>BCV</td>
<td>EMC</td>
<td>5264</td>
<td>50034321</td>
<td>4418880</td>
</tr>
<tr>
<td>/dev/rdsk/c4t1d1</td>
<td>BCV</td>
<td>EMC</td>
<td>5264</td>
<td>50035321</td>
<td>4418880</td>
</tr>
<tr>
<td>/dev/rdsk/c4t1d0</td>
<td>GK</td>
<td>EMC</td>
<td>5264</td>
<td>50040321</td>
<td>2880</td>
</tr>
<tr>
<td>/dev/rdsk/c3t1d1</td>
<td>GK</td>
<td>EMC</td>
<td>5264</td>
<td>50041321</td>
<td>2880</td>
</tr>
<tr>
<td>/dev/rdsk/c3t1d0</td>
<td>BCV</td>
<td>EMC</td>
<td>5264</td>
<td>50030161</td>
<td>4418880</td>
</tr>
<tr>
<td>/dev/rdsk/c3t1d1</td>
<td>BCV</td>
<td>EMC</td>
<td>5264</td>
<td>50031161</td>
<td>4418880</td>
</tr>
<tr>
<td>/dev/rdsk/c3t3d0</td>
<td>R2</td>
<td>EMC</td>
<td>5264</td>
<td>50004161</td>
<td>4418880</td>
</tr>
<tr>
<td>/dev/rdsk/c3t3d1</td>
<td>R2</td>
<td>EMC</td>
<td>5264</td>
<td>50005161</td>
<td>4418880</td>
</tr>
<tr>
<td>/dev/rdsk/c3t3d2</td>
<td>R1</td>
<td>EMC</td>
<td>5264</td>
<td>50006161</td>
<td>4418880</td>
</tr>
</tbody>
</table>
Preparing a Cluster for MetroCluster/SRDF

HP-UX device names to the actual devices in the Symmetrix configuration downloaded by EMC support staff. This number is further explained in Figure 4-14.

**Figure 4-14 Parsing the Symmetrix Serial Number**

- The Symmetrix ID is the same as the last two digits of the serial number of the Symmetrix frame, in this example 50.
- The next three hexadecimal digits are the unique Symmetrix device number that is seen in the output of the status command:

```
symrdf -g symdevgrpname query
```

and is used by the MetroCluster with EMC SRDF control script and saved in the file

```
/etc/cmcluster/package_name/symrdf.out
```

The contents of this file may be useful for debugging purposes.
- The next three digits indicate the Symmetrix host adapter (SA or FA) and port numbers; this is useful to see multiple host links to the same Symmetrix device. For example, PV links will show up as two HP-UX device file names with the same device number, but with different host adapter and port numbers.

3. Use the `symrdf` command on each Symmetrix disk array (that is, from both the R1 and the R2 side) to pair the logical device names for the R1 and R2 sides of each SRDF link:

```
# symrdf list
```

Sample output is shown in Figure 4-15 and Figure 4-16.
4. Match the logical device numbers in the `symrdf` listings with the HP-UX device file names in the output from the `syminq` command to see which devices are seen from each node to make sure that this node can see all necessary devices.

Use the Symmetrix ID to determine which Symmetrix array is connected to the node. Then use the Symmetrix device number to determine which devices are the same logical device seen by each node that is connected to the same Symmetrix unit. Record the
Building a Metropolitan Cluster Using MetroCluster/SRDF
Preparing a Cluster for MetroCluster/SRDF

HP-UX device file names in your table.

Table 4-6 show a partial mapping for a 4 node cluster connected to two Symmetrix arrays (95 and 50). Note that you may have many R1 and R2 devices and many gatekeepers for each package, so this table will be much larger for most clusters. Also, with M by N configurations, the number of devices increases according to the number of Symmetrix frames.

Table 4-6  Symmetrix Device and HP-UX Device Correlation

<table>
<thead>
<tr>
<th>Symmetrix ID, device #, and type</th>
<th>Node 1 /dev/rdsk device file name</th>
<th>Node 2 /dev/rdsk device file name</th>
<th>Node 3 /dev/rdsk device file name</th>
<th>Nodes 4 /dev/rdsk device file name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID 95  Dev# 005  Type R1</td>
<td>c0t4d0</td>
<td>c6t0d0</td>
<td>c4t0d0</td>
<td>c0t4d0</td>
</tr>
<tr>
<td>ID 50  Dev# 014  Type R2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ID 95  Dev# 00A  Type R2</td>
<td>c0t2d2</td>
<td>c0t4d2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ID 50  Dev# 012  Type R1</td>
<td></td>
<td></td>
<td>c3t0d2</td>
<td></td>
</tr>
<tr>
<td>ID 95  Dev# 040  Type GK</td>
<td>c0t15d0</td>
<td>c0t15d0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ID 50  Dev# 041</td>
<td></td>
<td></td>
<td>c3t15d1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>c5t15d1</td>
</tr>
</tbody>
</table>
NOTE

The Symmetrix device number may be the same or different in each of the Symmetrix units for the same logical device. In other words, the device number for the logical device on the R1 side of the SRDF link may be different from the device number for the logical device on the R2 side of the SRDF link.

The Symmetrix logical device numbers in these examples were configured to be the same number so that the cluster is easier to manage. If you are reconfiguring an existing cluster, the Dev and RDev devices will probably not be the same number.

When you are determining the configuration for the Symmetrix devices for a new installation, it is recommended that you try to use the same Symmetrix device number for the R1 devices and the R2 devices. It is also recommended that the same target and LUN number be configured for all nodes that have access to the same Symmetrix logical device.
Setting up 1 by 1 Configurations

The most common Symmetrix configuration used with MetroCluster/SRDF is a 1 by 1 configuration in which there is a single Symmetrix frame at each Data Center. This section describes how to set up this configuration using SymCLI and HP-UX commands. It is assumed that you have already set up the Symmetrix CLI database on each node as described in the previous section, “Preparing a Cluster for MetroCluster/SRDF.”

A basic 1 by 1 configuration is shown in Figure 4-17, which is a graphical view of the data in Table 4-6.

Creating Symmetrix Device Groups

A single Symmetrix device group must be defined for each package on each node that is connected to the Symmetrix. The following procedure must be done on each node that may potentially run the package:

The sample scripts `mk3symgrps nodename` can be modified to automate these steps.
Building a Metropolitan Cluster Using MetroCluster/SRDF

Setting up 1 by 1 Configurations

1. Use the `symdg` command, or modify the `mk3symgrps.nodename` script to define an R1 and an R2 device group for each package:

   ```
   # symdg create -type RDF1 devgroupname1
   Issue this command on nodes attached to the R1 side.
   # symdg create -type RDF2 devgroupname2
   Issue this command on nodes attached to the R2 side.
   ``

   The group name must be the same on each node on the R1 side. A different group name is used on each node on the R2 side. These group names are later placed in the shell script variables `PKGR1SYMGRP` and `PKGR2SYMGRP`.

2. Use the `symld` command to add all LUNs that comprise the Volume Group for that package on that host. The HP-UX device file names for all Volume Groups that belong to the package must be defined in one Symmetrix device group on the R1 side and one Symmetrix device group on the R2 side. All devices belonging to Volume Groups that are owned by an application package must be added to a single Symmetrix device group:

   ```
   # symld -g devgroupname1 add dev devnumber1
   # symld -g devgroupname1 add dev devnumber2
   ``

   This is where Table 4-6 on page 110 is helpful. Although the name of the device group must be the same on each node, the HP-UX device file names specified may be different on each node.

   When creating the Symmetrix device groups, you may specify only one HP-UX path to a particular Symmetrix device. Do not specify alternate paths (PVLinks). The SymCLI uses the HP-UX path only to determine to which Symmetrix device you are referring. The Symmetrix device may be added to the device group only once.

   **NOTE**

   Symmetrix Logical Device names must be the default names of the form `DEVnmm` (e.g., `DEV001`). Do not use the option for creating your own device names.

   The script must be customized for each system including:
   - particular HP-UX device file names
Building a Metropolitan Cluster Using MetroCluster/SRDF

Setting up 1 by 1 Configurations

- Symmetrix device group name (arbitrary, but unique name may be chosen for each group that defines all of the volume groups (VGs) belonging to a particular MC/ServiceGuard package).
- keyword RDF1 or RDF2

**Configuring Gatekeeper Devices**

**NOTE**
The sample scripts `mk4gatekpr.nodename` can be modified to automate these steps.

Gatekeeper devices must be unique per MC/ServiceGuard package to prevent contention in the Symmetrix when commands are issued, such as two or more packages starting up at the same time. Gatekeeper devices are unique to a Symmetrix unit. They are not replicated across the SRDF link. Gatekeeper devices are marked $GK$ in the `syminq` output, and are usually 2880 KB in size.

1. Define at least two gatekeepers per package per node (assuming PV links are used). They will only be available for use by that node. Each gatekeeper device is configured on different physical links:

   ```
   # symgate -sid sidnumber1 define dev devnumber1
   # symgate -sid sidnumber2 define dev devnumber2
   ```

2. Associate the gatekeeper devices with the Symmetrix device group for that package:

   ```
   # symgate -sid sidnumber1 -g devgroupname1 \ 
   associate dev devnumber1
   # symgate -sid sidnumber2 -g devgroupname1 \ 
   associate dev devnumber2
   ```

3. Define a pool of four or more additional gatekeeper devices that are not associated with any particular node. The SymCLI will switch to an alternate gatekeeper device if the path to the primary gatekeeper device fails. Certain software requires the existence of gatekeepers: OSM graphical interface and SymCLI. These gatekeeper devices are seen from all of the nodes.
Verifying the EMC Symmetrix Configuration

When you are finished with all these steps, use the `symrdf list` command to get a listing of all devices and their states.

You may want to back up the SymCLI database on each node so that you do not have to do these configuration steps again if a failure corrupts the database. The SymCLI database is a binary file in the directory `/var/symapi/db`.

Creating and Exporting Volume Groups

Use the following procedure to create volume groups and export them for access by other nodes. The sample script `mk1VGs` in the `Samples` directory can be modified to automate these steps.

1. Define the appropriate Volume Groups on each node that might run the application package. Use the commands:

   ```
   # mkdir /dev/vg xx
   # mknod /dev/vg xx /group c 64 0xnn0000
   ```

   where the name `/dev/vg xx` and the number `nn` are unique within the cluster.

2. Create volume groups only on the primary system. Use the `vgcreate` and the `vgextend` command, specifying the appropriate HP-UX device file names.

3. Use the `vgexport` command with the `-p` option to export the VGs on the primary system without removing the HP-UX device files:

   ```
   # vgchange -a n vgname
   # vgexport -v -s -p -m mapfilename vgname
   ```

   Make sure that you copy the map files to all of the nodes. The sample script `Samples/ftpit` shows a semi-automated way (using `ftp`) to copy the files. You need only enter the password interactively.

Importing Volume Groups on Other Nodes

Use the following procedure to import volume groups. The sample script `mk2imports` can be modified to automate these steps.

1. Import the VGs on all of the other systems that might run the
MC/ServiceGuard package and backup the LVM configuration. Make sure that you split the logical SRDF links before importing the VGs, especially if you are importing the VGs on the R2 side. Use the commands:

```
# symrdf -g symdevgrpname split -v
# vgimport -v -s -m mapfilename vgname
```

2. Back up the configuration. Use the following commands:

```
# vgchange -a y vgname
# vgcfgbackup vgname
# vgchange -a n vgname
# symrdf -g symdevgrpname establish -v
```

See the sample script `Samples/mk2imports`.

---

**NOTE**

Exclusive activation must be used for all volume groups associated with packages that use the EMC. The design of MetroCluster/SRDF assumes that only one system in the duster will have a VG activated at a time.

---

### Configuring PV Links

The examples in the previous sections show the use of the vgimport and vgexport commands with the `-s` option. Also, the mk1VGs script uses a `-s` in the vgexport command, and the mk2imports script uses a `-s` in the vgimport command. You may wish to remove this option from both commands if you are using PV links. The `-s` option to the vgexport command saves the volume group id (VGID) in the map file, but it does not preserve the order of PV links. To specify the exact order of PV links, do not use the `-s` option with vgexport, and in the vgimport command, enter the individual links in the desired order, as in the following example:

```
# vgimport -v -m mapfilename vgname linkname1 linkname2
```
Setting up M by N Configurations

Metropolitan clusters using EMC/SRDF can be built in configurations that use more than two EMC Symmetrix disk arrays. In such configurations, M arrays located in Data Center A may be connected to N arrays located in Data Center B. This section describes how to set up this configuration using SymCLI and HP-UX commands. It is assumed that you have already set up the Symmetrix CLI database on each node as described in the previous section, “Preparing a Cluster for MetroCluster/SRDF.” It is also assumed that Symmetrix PowerPath software is installed on all nodes.

Figure 4-18 shows a 2 by 2 configuration. Data in this figure are used in the example commands given in the following sections. The example shows R1 devices at one data center and R2 devices with Business Continuity Volumes (BCVs) at the other, but a bidirectional configuration is also possible, with R1 devices on both sites.

Figure 4-18 Devices and Symmetrix Units in M by N Configurations
Creating Symmetrix Device Groups

For each node on the R1 side (node1 and node2), create the device groups as follows. You have to create two device groups because device groups do not span frames.

**NOTE**

In a 1 by 2 or 2 by 1 configuration, on either the R1 or R2 side, whichever side has the one Symmetrix frame, a minimum of two device groups is required. The rule of thumb is that you need one device group per RDF group (RA group), even if you have only one frame on that side.

The following examples are based on the 2 by 2 configuration shown in Figure 4-18.

Create device groups using the following commands on each node on the R1 side:

```bash
# symdg -type RDF1 create dgoraA
# symdg -type RDF1 create dgoraB
```

For each node on the R2 side (node3 and node4), create the device groups as follows. Note that you have to create two device groups because device groups do not span frames. Do the following on each node on the R2 side:

```bash
# symdg -type RDF2 create dgoraA
# symdg -type RDF2 create dgoraB
```

For each node on the R1 side (node1 and node2), assign the R1 devices to the device groups as follows:

```bash
# symld -sid 638 -g dgoraA add dev 00C
# symld -sid 638 -g dgoraA add dev 00D
# symld -sid 130 -g dgoraB add dev 010
# symld -sid 130 -g dgoraB add dev 011
```

For each node on the R2 side (node3 and node4), assign the R2 devices to the device groups as follows:

```bash
# symld -sid 021 -g dgoraA add dev 018
# symld -sid 021 -g dgoraA add dev 019
# symld -sid 363 -g dgoraB add dev 050
```
Building a Metropolitan Cluster Using MetroCluster/SRDF

Setting up M by N Configurations

# symld -sid 363 -g dgoraB add dev 051

On each node on the R2 side (node3 and node4), associate the local BCV devices to the R2 device group:

# symbcv -g dgoraA add dev 01A
# symbcv -g dgoraA add dev 01B
# symbcv -d dgoraB add dev 052
# symbcv -d dgoraB add dev 053

To manage the BCV devices from the R1 side, you need to associate the BCV devices with the device groups that are configured on the R1 side. Use the following commands on hosts directly connected to the R1 Symmetrix:

# symbcv -g dgoraA associate dev 01A -rdf
# symbcv -g dgoraA associate dev 01B -rdf
# symbcv -g dgoraB associate dev 052 -rdf
# symbcv -g dgoraB associate dev 053 -rdf

Now establish the BCV devices using the following commands from the R2 side:

# symmir -g dgoraA -full est
# symmir -g dgoraB -full est

Alternatively, you can establish the BCV devices with the following commands from the R1 side:

# symmir -g dgoraA -full est -rdf
# symmir -g dgoraB -full est -rdf

Configuring Gatekeeper Devices

You need a gatekeeper device for each device group in the consistency group that will be built in a later step. Use the following commands on all nodes on the R1 side to define gatekeepers and associate them with device groups:

# symgate -sid 638 define dev 010
# symgate -sid 130 define dev 009
# symgate -sid 638 -g dgoraA associate dev 010
Building a Metropolitan Cluster Using MetroCluster/SRDF

Setting up M by N Configurations

Use the following commands on all nodes on the R2 side to define gatekeepers and associate them with device groups:

```bash
#symgate -sid 130 -g dgoraB associate dev 009
#symgate -sid 021 define dev 002
#symgate -sid 363 define dev 00B
#symgate -sid 021 -g dgoraA associate dev 002
#symgate -sid 363 -g dgoraB associate dev 00B
```

Defining the Consistency Groups

To configure consistency groups for use with EMC/SRDF, you first create device groups and gatekeeper groups as described in previous sections. Then you create the consistency groups and add the devices to them as well.

**NOTE**

Each package requires its own unique consistency group name.

Use the following steps for each package:

1. On each node in the cluster, create an empty consistency group using the symcg command:
   ```bash
   #symcg create cgoradb
   ```
   You can use the same name on all nodes.

2. Add each device that is going to be used in the consistency group. Use the appropriate SID numbers and device names for the data center that the node is a part of. For example, on node 1 and node 2 in Data Center A:
   ```bash
   #symcg -cg cgoradb -sid 638 add dev 00C
   #symcg -cg cgoradb -sid 638 add dev 00D
   #symcg -cg cgoradb -sid 130 add dev 010
   #symcg -cg cgoradb -sid 130 add dev 011
   ```
   And on node 3 and node 4 in Data Center B:
   ```bash
   #symcg -cg cgoradb -sid 021 add dev 018
   ```
Building a Metropolitan Cluster Using MetroCluster/SRDF

Setting up M by N Configurations

3. Enable the consistency group:

```
# symcg -cg cgoradb -sid 021 add dev 019
# symcg -cg cgoradb -sid 363 add dev 050
# symcg -cg cgoradb -sid 363 add dev 051
```

**NOTE**

This important step must be carried out on every node.

```
# symcg -g cgoradb enable
```

4. Establish the BCV devices in the secondary Symmetrix as a mirror of the standard device. From either node3 or node4, issue the following commands:

```
# symmir -g dgoraA -full est
# symmir -g dgoraB -full est
```

Alternatively, from either node1 or node2, issue the following commands:

```
# symmir -g dgoraA -full est -rdf
# symmir -g dgoraB -full est -rdf
```

### Creating Volume Groups

The following procedures assume you are creating volume groups for a cluster with a 2 by 2 Symmetrix configuration like that of Figure 4-18. Use the following steps on node1:

1. Create the physical volumes:

```
# pvcreate -f /dev/rdsk/c6t0d0
# pvcreate -f /dev/rdsk/c6t0d1
# pvcreate -f /dev/rdsk/c5t0d2
# pvcreate -f /dev/rdsk/c5t0d3
```

2. Create the directories and special files for the volume groups:

```
# mkdir /dev/vgoraA
# mkdir /dev/vgoraB
```
Building a Metropolitan Cluster Using MetroCluster/SRDF

Setting up M by N Configurations

```bash
# mknod /dev/vgoraA/group c 64 0x01000
# mknod /dev/vgoraB/group c 64 0x02000

3. Create the volume groups. Be careful not to span Symmetrix frames:
   # vgcreate /dev/vgoraA /dev/rdsk/c6t0d0
   # vgextend /dev/vgoraA /dev/rdsk/c6t0d1
   # vgcreate /dev/vgoraB /dev/rdsk/c5t0d2
   # vgextend /dev/vgoraB /dev/rdsk/c5t0d3

4. Create the logical volumes. (XXXX indicates size in MB)
   # lvcreate -L XXXX /dev/vgoraA
   # lvcreate -L XXXX /dev/vgoraB

5. Install a VxFS file system on the logical volumes:
   # newfs -F vxfs /dev/vgoraA/rlvol1
   # newfs -F vxfs /dev/vgoraB/rlvol1

6. Create map files to permit exporting the volume groups to other systems:
   # vgchange -a n vgoraA
   # vgchange -a n vgoraB
   # vgexport -v -s -p -m /tmp/vgoraA.map vgoraA
   # vgexport -v -s -p -m /tmp/vgoraB.map vgoraB

7. Copy the map files to the other nodes in the cluster:
   # rcp /tmp/vgoraA.map node2:/tmp/vgoraA.map
   # rcp /tmp/vgoraB.map node2:/tmp/vgoraB.map

8. Split the SRDF logical links:
   # symrdf -g dgoraA split -v
   # symrdf -g dgoraB split -v

On node2, node3, and node4, perform the following steps:

1. Create the volume group directories and special files:
   # mkdir /dev/vgoraA
   # mkdir /dev/vgoraB
```
Configuring MC/ServiceGuard Packages for Automatic Disaster Recovery

Before you can implement these procedures you must:

- Configure your cluster according to disaster tolerant architecture guidelines outlined in “Designing a Disaster Tolerant Architecture for use with MetroCluster/SRDF” on page 90.
- Configure the MC/ServiceGuard cluster according to the procedures outlined in Managing MC/ServiceGuard (part number B3936-90024).
- Create the SymCLI database, and build Symmetrix device groups, consistency groups, and gatekeepers for each package. Export exclusive volume groups for each package as described in “Preparing a Cluster for MetroCluster/SRDF” on page 100. This must be done on each node that will potentially run the package.
- Install the MetroCluster with EMC SRDF product on all nodes according to the instructions in the MetroCluster with EMC SRDF Release Notes.

When you have completed these steps, packages will be able to automatically fail over to an alternate node in another data center and still have access to the data it needs to function.

This procedure must be repeated on all the cluster nodes for each MC/ServiceGuard application package so the application can failover to any of the nodes in the cluster. Customizations include setting environment variables and supplying customer-defined run and halt commands, as appropriate. The package control script must also be customized for the particular application software that it will control. Consult Managing MC/ServiceGuard for more detailed instructions on how to start, halt, and move packages and their services between nodes in a cluster.
Building a Metropolitan Cluster Using MetroCluster/SRDF

Configuring MC/ServiceGuard Packages for Automatic Disaster Recovery

For ease of troubleshooting, you may want to configure and test one package at a time.

1. Create a directory `/etc/cmcluster/package_name` for each package and copy `/opt/cmcluster/toolkits/SGSRDF/srdfpkg.cntl` to the package directory:

   ```bash
   # cp /opt/cmcluster/toolkits/SGSRDF/srdfpkg.cntl
   /etc/cmcluster/package_name
   ```

   If you haven't already, copy the scripts in the `Samples` directory to another directory (e.g., `/etc/cmcluster/SGSRDF`). Change the name of `srdfpkg.cntl` to `package_name.cntl`.

2. Create an MC/ServiceGuard package configuration file with the commands:

   ```bash
   # cd /etc/cmcluster/package_name
   cmmakepkg -p package_name.ascii
   ```

   Each package should have its own directory and uniquely named control script.

   **NOTE**

   Do not create an MC/ServiceGuard package control script template with the `cmmakepkg -s` command or by using SAM. The package control script template included with this product (`/opt/cmcluster/toolkits/SGSRDF/srdfpkg.cntl`) should be copied to the package directory and used instead.

3. In the `package_name.ascii` file, list the node names in the order in which you want the package to fail over. It is recommended for performance reasons, that you have the package fail over locally first, then to the remote data center. Be sure to set the `MAX_CONFIGURED_PACKAGES` parameter to 1 or more, depending on the number of packages that will run on the cluster.

   This toolkit may increase package startup time by 5 minutes or more. Packages with many disk devices will take longer to start up than those with fewer devices due to the time needed to get device status from the Symmetrix. Clusters with multiple packages that use devices on the Symmetrix will all cause package startup time to increase when more than one package is starting at the same time.

   The value of `RUN_SCRIPT_TIMEOUT` in the package ASCII file should
Building a Metropolitan Cluster Using MetroCluster/SRDF

Configuring MC/ServiceGuard Packages for Automatic Disaster Recovery

be set to NO_TIMEOUT or to a large enough value to take into consideration the extra startup time due to getting status from the Symmetrix.

4. Customize the control script as appropriate to your application using the guidelines in Managing MC/ServiceGuard. Standard MC/ServiceGuard package customizations include modifying the VG, LV, FS, IP, SUBNET, SERVICE_NAME, SERVICE_CMD and SERVICE_RESTART parameters. Be sure to set LV_UMOUNT_COUNT to 1 or greater.

5. Further customize the control script for use with EMC SRDF as follows:

a. Add the path where the SymCLI software binaries have been installed to the PATH environment variable. If the software is in the usual location, /usr/symcli/bin, you can just uncomment the line in the script.

b. Uncomment AUTOR1RWNL, AUTOR1UIP, AUTOR2RWNL and AUTOR2WDNL environment variables. It is recommended that you retain the default values of these variables unless you have a specific business requirement to change them. See Appendix A for an explanation of these variables.

c. Uncomment the PKGDIR variable and set it to the full path name of the directory where the control script has been placed. This directory must be unique for each package and is used for status data files. For example, set PKGDIR to /etc/cmcluster/package_name, removing any quotes around the file names.

d. Uncomment the PKGR1SYMGRP and PKGR2SYMGRP variables and set them to the Symmetrix device group names for local and remote EMC Symmetrix disk arrays given in the symdg list command.

e. Uncomment the R1HOST array variable and set the elements to the node names of the systems on the R1 side of the SRDF link.

NOTE
If you are using an M by N configuration, configure these variables PKGR1SYMGRP and PKGR2SYMGRP with the names of the consistency groups.

Chapter 4 125
Building a Metropolitan Cluster Using MetroCluster/SRDF

Configuring MC/ServiceGuard Packages for Automatic Disaster Recovery

The order of the node names is not important.

f. Uncomment the R2HOST array variable and set the elements to the node names of the systems on the R2 side of the SRDF link. The order of the node names is not important.

g. Uncomment the RETRY and RETRYTIME variables. These variables are used to decide how often and how many times to retry the Symmetrix status commands. The defaults should be used for the first package. For other packages RETRYTIME should be altered to avoid contention when more than one package is starting on a node. RETRY * RETRYTIME should be approximately five minutes to keep package startup time under 5 minutes.

<table>
<thead>
<tr>
<th>RETRYTIME</th>
<th>RETRY</th>
</tr>
</thead>
<tbody>
<tr>
<td>pkgA</td>
<td>60 seconds</td>
</tr>
<tr>
<td>pkgB</td>
<td>43 seconds</td>
</tr>
<tr>
<td>pkgC</td>
<td>33 seconds</td>
</tr>
</tbody>
</table>

h. Uncomment the CLUSTERTYPE variable and set it to METRO. (The value CONTINENTAL is only for use with the ContinentalClusters product, described in Chapter 5.)

i. If you are using an M by N configuration, be sure that the variable CONSISTENCYGROUPS is set to 1 in the control script:

   CONSISTENCYGROUPS=1

6. Add customer-defined run and halt commands in the appropriate places by modifying the RUN_SCRIPT and HALT_SCRIPT parameters according to the needs of the application. Include the full pathname of the control script:

   /etc/cmcluster/package_name/package_name.cntl.

   See Managing MC/ServiceGuard for more information on these functions.

7. Distribute MetroCluster with EMC SRDF control script files to other nodes in the cluster by using ftp or rcp:

   # rcp -p /etc/cmcluster/package_name/package_name.cntl /other_node:/etc/cmcluster/package_name/package_name.cntl

   See the example script ftpit to see how to semi-automate the copy
Building a Metropolitan Cluster Using MetroCluster/SRDF

Configuring MC/ServiceGuard Packages for Automatic Disaster Recovery

using ftp. This script assumes the package directories already exist on all nodes.

8. Split the SRDF logical links for the disks associated with the application package. See the script pre.cmquery for an example of how to automate this task. The script must be customized with the Symmetrix device group names.

When using SRDF, the devices on the R2 side of the SRDF link are normally read-only. Before running cmquerycl and cmapplyconf, you must first split the SRDF logical link for all disk devices that are to be configured in MC/ServiceGuard packages. The links are then re-established once the new configuration has been applied.

9. Check the configuration using the cmcheckconf -P package_name.ascii, then apply the MC/ServiceGuard configuration using the cmapplyconf -P package_name.ascii command or SAM.

10. Restore the SRDF logical links for the disks associated with the application package. See the script Samples/post.cmapply for an example of how to automate this task. The script must be customized with the Symmetrix device group names. You may want to redirect the output of this script to a file for debugging purposes.

11. Verify that each node in the MC/ServiceGuard cluster has the following files in the directory /etc/cmcluster/package_name:

    package_name.cntl  (MC/ServiceGuard package control script)
    package_name.ascii (MC/ServiceGuard package ASCII config file)

    Any other scripts you use to manage MC/ServiceGuard packages.

The cluster is ready to switch packages automatically to nodes in remote data centers using MetroCluster/SRDF.
Maintaining a Cluster that Uses MetroCluster/SRDF

While the cluster is running, all EMC Symmetrix disk arrays that belong to the same MC/ServiceGuard package, and are defined in a single SRDF group must be in the same state at the same time. Manual changes of these states can cause the package to halt due to unexpected conditions. In general, it is recommended that no manual change of states be performed while the package and the cluster are running.

There might be situations when the package has to be taken down for maintenance purposes without having the package move to another node.

The following procedure is recommended for normal maintenance of the MetroCluster with EMC SRDF:

1. Stop the package with the appropriate MC/ServiceGuard command.
   
   `# cmhaltpkg pkgname`

2. Split the logical SRDF links for the package.
   
   `# Samples/pre.cmquery`

3. Distribute the MetroCluster with EMC SRDF configuration changes.
   
   `# cmapplyconf -P pkgconfig`

4. Restore the logical SRDF links for the package.
   
   `# Samples/post.cmapply`

5. Start the package with the appropriate MC/ServiceGuard command:
   
   `# cmmodpkg -e pkgname`

No checking of the status of the SA/FA ports is done. It is assumed that at least one PVL link is functional. Otherwise, the VG activation will fail.

Planned maintenance is treated the same as a failure by the cluster. If you take a node down for maintenance, package failover and quorum calculation is based on the remaining nodes. Make sure that nodes are taken down evenly at each site, and that enough nodes remain on-line to form a quorum if a failure occurs. See “Example Failover Scenarios with Two Arbitrators” on page 96.
Managing Business Continuity Volumes

The use of Business Continuity Volumes is recommended with all implementations of MetroCluster/SRDF, and it is required with M by N configurations, which employ consistency groups. These BCV devices will provide a good copy of the data when it is necessary to recover from a rolling disaster—a second failure that occurs while we are attempting to recover from the first failure.

Protecting against Rolling Disasters

An example of a rolling disaster with MetroCluster/SRDF is as follows. At time $T_0$, all the SRDF links go down. The application continues to run on the R1 side. At time $T_1$, the SRDF links are restored, and at $T_2$ a manual resynchronization is started to resync new data from the R1 to the R2 side. At time $T_3$, while resynchronization is in progress, the R1 site fails, and the application starts up on the R2 side. Since the resynchronization did not complete when there was a failure on the R1 side, the data on the R2 side is corrupt.

Using the BCV in Resynchronization

In general, you use the business continuity volumes only in cases where a resynchronization will take place. First, you split off a consistent copy of the data, then perform the resynchronization, then re-establish the BCV mirroring. To perform these steps, use the following sequence of commands:

1. Split the BCV in the secondary Symmetrix from the mirror group to save a good copy of the data. From node3 or node4:
   
   ```
   # symmir -g dgoraA split
   # symmir -g dgoraB split
   ``
   Alternatively, from node1 or node2:
   
   ```
   # symmir -g dgoraA split -rdf
   # symmir -g dgoraB split -rdf
   ```

2. Resync the data from R1 to R2 devices:
   
   ```
   # symrdf -cg cgoradb est
   ```
3. Use the following command to monitor the status of the RDF pair state between the R1 and R2 devices:

```
# symrdf -cg cgoradb query
```

Once the column with the “RDF Pair STATE” shows the state as “Synchronized” for all the devices, you can proceed to the next step.

4. Re-establish the BCV devices in the secondary Symmetrix as a mirror of the standard device. From node 3 or node4:

```
# symmir -g dgoraA -full est
# symmir -g dgoraB -full est
```

Alternatively, from node1 or node2:

```
# symmir -g dgoraA -full est -rdf
# symmir -g dgoraB -full est -rdf
```
Building a Continental Cluster

Unlike metropolitan and campus clusters, which have a single-cluster architecture, a continental cluster uses multiple MC/ServiceGuard clusters to provide application recovery over wide areas. Using the ContinentalClusters product, two independently functioning clusters are set up in such a way that in the event of a disaster, one cluster can take over the critical operations formerly carried out by the other cluster.

Disaster tolerance is obtained by eliminating the cluster itself as a single point of failure. This chapter describes the configuration and management of a basic continental cluster through the following topics:

- Understanding Continental Cluster Concepts
- Designing a Disaster Tolerant Architecture for use with ContinentalClusters
- Preparing the Clusters
- Building the ContinentalClusters Configuration
- Testing the Continental Cluster
- Switching to the Recovery Packages in Case of Disaster
- Restoring Disaster Tolerance
- Maintaining a Continental Cluster

For a description of the cascading failover configuration, see Chapter 8. Refer to Appendixes C and D for additional information on the ContinentalClusters command set and on configuration file parameters.

NOTE
This chapter only briefly addresses data replication, highly available WANs, and site security and communication. Chapters 6 and 7 give details on physical data replication using the HP SureStore E Disk Array XP Series with Continuous Access XP and the EMC Symmetrix with the SRDF facility. Information on logical data replication via the Oracle Standby Database is available from your HP representative. Also see the “White Papers” area on web page http://docs.hp.com/hpux/ha.
Building a Continental Cluster

Understanding Continental Cluster Concepts

The ContinentalClusters product provides the ability to monitor a high availability cluster and fail over mission critical applications to another cluster if the monitored cluster should become unavailable. In the following example, the Los Angeles cluster runs the mission critical application and replicates data to the New York cluster, which has another copy of the mission critical application ready to run in case of failover. In addition, ContinentalClusters supports mutual recovery, which allows for mission critical applications to be run on each cluster, with each cluster configured to recover the mission critical applications of the other.

Because clusters may be separated over wide geographical distances, and because they have independent function, the operation of clusters in a ContinentalClusters configuration is somewhat different from that of typical MC/ServiceGuard clusters. A typical ContinentalClusters environment is shown in Figure 5-1.
Two packages are running on the cluster in Los Angeles, and their data is replicated to the cluster in New York. Physical data replication is carried out using ESCON (Enterprise Storage Connect) links between the disk array hardware in New York and Los Angeles via an ESCON/WAN converter at each end. The New York cluster is running a monitor that checks the status of the Los Angeles cluster. In this example, the Los Angeles cluster runs just like any ServiceGuard cluster, with applications configured in packages that may fail from node to node as necessary. The New York cluster is configured with a recovery version of the packages that are running on the Los Angeles cluster. These packages do not run under normal circumstances, but are set to start up when they are needed. In addition, either cluster may run other packages that are not involved in Continental Clusters operation.

**Mutual Recovery Configuration**

Bi-directional failover is now supported in what is called a mutual recovery configuration. This lets you define recovery groups for primary packages running in both component clusters in the Continental Clusters configuration. Figure 5-2 shows a mutual recovery configuration.
Building a Continental Cluster

Understanding Continental Cluster Concepts

In the above figure, the salespkg is running on the New York cluster and can be recovered by the Los Angeles cluster. Similarly, the custpkg running on the Los Angeles cluster can be recovered by the New York cluster. As stated previously, physical data replication is carried out using ESCON (Enterprise Storage Connect) links between the disk array hardware in New York and Los Angeles via an ESCON/WAN converter at each end. Each cluster is running a monitor that checks the status of the alternate cluster.

As shown in the above example, each cluster runs just like any ServiceGuard cluster, with applications configured in packages that may fail from node to node as necessary. Each cluster is configured with a recovery version of the packages that are running on the alternate cluster. These packages do not run under normal circumstances, but are set to start up when they are needed. In addition, either cluster may run other packages that are not involved in ContinentalClusters operation.

Application Recovery in a Continental Cluster

If a given cluster of a Continental cluster should become unavailable, ContinentalClusters allows an administrator to issue a single command (cmrecovercl, described later) to transfer mission critical applications from the that cluster to another cluster, making sure that the packages do not run on both clusters at the same time. Transfer is not automatic, although it is automated through a recovery command, which a root user must issue. The result after issuing the recovery command is shown in Figure 5-3.
The movement of an application from one cluster to another cluster does not replace local failover activity; packages are normally configured to fail over from node to node as they would on any high availability cluster. **Cluster recovery**—failover of packages to a different cluster—occurs only after the following:

- ContinentalClusters detects the problem.
- ContinentalClusters sends you a notification of the problem.
- You verify that the monitored cluster has failed.
- You issue the cluster recovery command.

**Monitoring over a Wide Area Network**

A monitor package running on one cluster tracks the health another cluster and sends notification to system administrators if the state of the monitored cluster changes. (If a cluster contains any recovery packages it must be monitored.) The monitor software polls the monitored cluster at
Understanding Continental Cluster Concepts

A specific MONITOR_INTERVAL defined in an ASCII configuration file, which also indicates when and where to send messages if there is a state change.

The physical separation between clusters will require communication by way of a Wide Area Network (WAN). Since the polling takes place across the WAN, interruptions of WAN service cannot always be differentiated from cluster failure states. This means that if the WAN is unreliable, the monitoring facility will often detect and report an unreachable state for the monitored cluster that is actually an interruption of WAN service.

Because the monitoring is indeterminate in some instances, information from independent sources must be gathered to determine the need for proceeding with the recovery process. For these reasons, cluster recovery is not automatic, but must be initiated by a root user. Once initiated, however, the cluster recovery is automated to reduce the chance of human error that might occur if manual steps were needed. In ContinentalClusters, a system of cluster events and notifications is provided so that events can be easily tracked, and so that users will know when to seek additional information before initiating recovery.

**Cluster Events**

A cluster event is a change of state in a monitored cluster. The four cluster states reported by the monitor are **Unreachable**, **Down**, **Up**, and **Error**. Table 5-1 summarizes possible causes for the cluster events with regard to both the monitored cluster and the WAN. It is clear that in many cases, the causes of cluster events are indeterminate without additional information that is not available to the software.

<table>
<thead>
<tr>
<th>Cluster Event (Old state -&gt; New state)</th>
<th>Cluster-related causes</th>
<th>WAN-related causes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Up -&gt; Unreachable</td>
<td>Cluster went down; no nodes are responding to network inquiries</td>
<td>WAN failure</td>
</tr>
<tr>
<td>Down -&gt; Unreachable</td>
<td>Cluster was down and nodes are no longer responding</td>
<td>WAN failure</td>
</tr>
<tr>
<td>Error -&gt; Unreachable</td>
<td>Error resolved but cluster down and nodes not responding; or WAN-related cause</td>
<td>WAN failure</td>
</tr>
</tbody>
</table>
**Table 5-1** Monitored States and Possible Causes

<table>
<thead>
<tr>
<th>Cluster Event (Old state -&gt; New state)</th>
<th>Cluster-related causes</th>
<th>WAN-related causes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Up -&gt; Down</td>
<td>Cluster has been halted, but at least one node is still responding to network inquiries</td>
<td>No WAN problems</td>
</tr>
<tr>
<td>Error -&gt; Down</td>
<td>Error resolved, cluster is down</td>
<td>WAN problem was fixed, cluster is down</td>
</tr>
<tr>
<td>Unreachable -&gt; Down</td>
<td>Cluster nodes were rebooted but the cluster was not started</td>
<td>WAN came up but the cluster was not running</td>
</tr>
<tr>
<td>Up -&gt; Error</td>
<td>ServiceGuard version or security file mismatch, software error</td>
<td>WAN is misconfigured, or DNS server crashed or set up incorrectly</td>
</tr>
<tr>
<td>Down -&gt; Error</td>
<td>ServiceGuard version or security file mismatch, software error</td>
<td>WAN is misconfigured, or DNS server crashed or set up incorrectly</td>
</tr>
<tr>
<td>Unreachable -&gt; Error</td>
<td>ServiceGuard version or security file mismatch, software error</td>
<td>WAN problem was fixed, but the error condition still exists</td>
</tr>
<tr>
<td>Down -&gt; Up</td>
<td>Cluster started</td>
<td>No WAN problems</td>
</tr>
<tr>
<td>Unreachable -&gt; Up</td>
<td>Cluster nodes were rebooted and the cluster started</td>
<td>WAN came up and the cluster was already running</td>
</tr>
<tr>
<td>Error -&gt; Up</td>
<td>Error resolved, cluster is up</td>
<td>WAN problem was fixed, cluster is up</td>
</tr>
</tbody>
</table>

---

**NOTE**

There is only one condition under which cmclsentryd will determine that the cluster has Error status: all nodes are unreachable except those which have ServiceGuard Error status. (If any nodes are Down or Up, then the cluster status will take one of those values, rather than Error.)
Interpreting the Significance of Cluster Events

Because some cluster events (e.g., Up -> Unreachable) can be caused by changes in either a cluster state or a WAN state, additional independent information is required to achieve the primary objective of determining whether you need to recover a cluster’s applications. Sources of independent information include:

- Contact with the WAN provider
- Contact with the administrator of the monitored cluster
- Contact with local cluster administrator
- Contact with company executives

When worrisome cluster events persist, you obtain as much information as possible, including authorization to recover, if your business practices require this, and then issue the recovery command.

How Notifications Work

A central part of the operation of Continental Clusters is the transmission of notifications following the detection of a cluster event. Notifications occur at specifically coded times, and at two different levels:

- **Alert**—when a cluster event should be considered noteworthy.
- **Alarm**—when an event shows evidence of a cluster failure.

Notifications are typically sent as:

- Email messages
- SNMP traps
- Text log files
- OPC messages to OpenView IT/Operations

In addition, notifications are sent to an event log on the system where monitoring is taking place.

---

**NOTE**

An email message can be sent to an address supplied by a pager service that will forward the message to a specified pager system. Contact your pager service provider for more information.
Building a Continental Cluster

Understanding Continental Cluster Concepts

Alerts
Alerts are intended as informational. Some typical uses of alerts include:

- Notification that a cluster has been halted for a significant amount of time.
- Notification that a cluster has come up after being down or unreachable.
- Notification that a cluster came down for any reason.
- Notification that a cluster has been in an unreachable state for a short period of time. An alert is sent in this case as a warning that an alarm might be issued later if the cluster's state remains unreachable for a longer time.

The expected process in dealing with alerts is to continue watching for additional notifications and to contact individuals at the site of the monitored cluster to see whether problems exist.

Alarms
Alarms are intended to indicate that a cluster failure might have taken place. The most common example of an alarm is the following:

- Notification that a cluster has been in an unreachable state for a significant amount of time that you specify.

The expected process in dealing with cluster events that persist at the alarm level is to obtain as much information as possible, including authorization to recover, if your business practices require this, and then to issue the recovery command.

Creating Notifications for Failure Events
For events that might indicate cluster failure, you can show the escalation of your concern over cluster health by defining alerts followed by one or more alarms. A typical sequence is to issue a cluster alert at 5 minutes and 10 minutes followed by a cluster alarm at 15 minutes. This could be accomplished by entering two CLUSTER_ALERT lines in the configuration file, and one CLUSTER_ALARM line. A detailed example is provided in the comments in the ASCII configuration file template, shown in “Editing Section 3—Monitoring Definitions” on page 170.
Creating Notifications for Events that Indicate a Return of Service

For those events that indicate that the cluster is back online or that communication with the monitor has been restored, use cluster alerts to show the de-escalation of concern. In this case, use a `CLUSTER_ALERT` line in the configuration file with a time of zero (0), so that notifications are sent as soon as the return to service is detected.

Performing Cluster Recovery

When a `CLUSTER_ALARM` is issued, there may be a need for recovery, and the recovery command, `cmrecovercl`, is enabled for use by the root user. Cluster recovery is carried out at the site of the recovery cluster by using the `cmrecovercl` command, as follows:

```
# cmrecovercl
```

This command will fail if a cluster alarm has not been issued. The command has the effect of halting any data replication activity from the failed cluster to the local cluster, and starting up on the local cluster all the recovery packages that are pre-configured in recovery groups, which are the units of recovery in a continental cluster. After the `cmrecovercl` command is issued, there is a delay of at least 90 seconds per recovery group as the command makes sure that the package is not active on another cluster.

Cluster recovery is done, in effect, as a last resort, after all other approaches to restore the unavailable cluster have been exhausted. It is important to remember that cluster recovery sets in motion a process that cannot be easily reversed. Unlike the failover of a package from one node to another, failing a package from one cluster to another normally involves a significant quantity of data that is being accessed from a new set of disks. Returning control to the original cluster will involve resynchronizing this data and resetting the roles of the clusters in a process that is easier for some data replication techniques than others.

---

**NOTE**

After a recovery, you cannot reverse directions and return a package to its original cluster without first reconfiguring the data replication hardware and/or software and synchronizing data. Therefore, you should be very cautious when deciding to use the `cmrecovercl` command.
Notes on Packages in a Continental Cluster

Packages have somewhat different behavior in a continental cluster than in a normal ServiceGuard environment. There are specific differences in:

- Startup and Switching Characteristics
- Network Attributes

Startup and Switching Characteristics

Normally, an application (package) can run on only one node at a time in a cluster. However, in a continental cluster, there are two clusters in which an application—the primary package or the recovery package—could operate on the same data. The primary package and the recovery package must not both be allowed to run at the same time. To prevent this, it is very important to ensure that packages are not allowed to start automatically and are not started up at inappropriate times.

To keep packages from starting up automatically when a cluster starts, you must set the AUTO_RUN (PKG_SWITCHING_ENABLED used prior to ServiceGuard 11.12) parameter for all primary and recovery packages to NO. Then use the cmmodpkg command with -e <packagename> option to start up only the primary packages and enable switching. The cmrecovercl command, when run, will start up the recovery packages and enable switching during the cluster recovery operation.

WARNING

After initial testing is complete, the cmrunpkg and cmmodpkg commands or the equivalent options in SAM should never be used to start a recovery package unless cluster recovery has already taken place.

To prevent packages from being started at the wrong time and in the wrong place, you can use the following strategies:

- Set the AUTO_RUN (PKG_SWITCHING_ENABLED used prior to ServiceGuard 11.12) parameter for all primary and recovery packages to NO.
- Ensure that recovery package names are well known, and that personnel understand they should never be started with a cmrunpkg or cmmodpkg command unless the cmrecovercl command has been invoked first.
Building a Continental Cluster

Understanding Continental Cluster Concepts

• Once the continental cluster is in production, create special scripts for the execution of the ServiceGuard commands cmrunpkg and cmmodpkg. A script could be written that would run the cmrunpkg or cmmodpkg command, but exit with an error or warning message if someone inadvertently tried to start a specified list of your recovery packages with these commands.

• If a cluster has no packages to run before recovery, then do not allow packages to be run on that cluster with SAM.

Network Attributes

Another important difference between the packages in a continental cluster and the packages configured in a standard ServiceGuard cluster is that different subnets are used in recovery packages than the subnets in the primary packages. The client application must be designed to reconnect to the appropriate IP address following a recovery operation.

How MC/ServiceGuard commands work in a Continental Cluster

ContinentalClusters packages are manipulated manually by the user via ServiceGuard commands and by cmd automatically in the same way as any other packages.

In a continental cluster the recovery package are not allowed to run at the same time as the primary, data sender, or data receiver packages. To enforce this, several MC/ServiceGuard commands behave in a slightly different manner when used in a continental cluster.

Table 5-2 describes the ServiceGuard commands whose behavior is different in a continental cluster environment. Specifically, when one of the following commands attempts to start or enable switching of a package, it first checks the status of the other packages in the recovery group. Based on this status, the operation is either allowed or disallowed.
### Table 5-2  ServiceGuard and ContinentalClusters Commands

<table>
<thead>
<tr>
<th>Commands</th>
<th>How the commands work in SG</th>
<th>How the commands work in ContinentalClusters</th>
</tr>
</thead>
<tbody>
<tr>
<td>cmrunpkg</td>
<td>runs a package</td>
<td>will not start a recovery package if any of the primary, data receiver, or data sender package is in the same recovery group is running or enabled. will not start a primary, data receiver, or data sender package if the recovery package in the same recovery group is running or enabled.</td>
</tr>
<tr>
<td>cmmodpkg -e</td>
<td>enable switching attribute for a highly available package</td>
<td>will not enable switching on a recovery package if any of the primary, data receiver, or data sender package is in the same recovery group is running or enabled, will not start a primary, data receiver, or data sender package if the recovery package in the same recovery group is running or enabled.</td>
</tr>
</tbody>
</table>
Building a Continental Cluster

Understanding Continental Cluster Concepts

Table 5-2  ServiceGuard and ContinentalClusters Commands

<table>
<thead>
<tr>
<th>Commands</th>
<th>How the commands work in SG</th>
<th>How the commands work in ContinentalClusters</th>
</tr>
</thead>
<tbody>
<tr>
<td>cmhaltnode -f</td>
<td>halts a node in a highly available cluster</td>
<td>will not re-enable switching on a recovery package if any of the primary, data receiver, or data sender package is in the same recovery group is running or enabled. will not start a primary, data receiver, or data sender package if the recovery package in the same recovery group is running or enabled</td>
</tr>
<tr>
<td>cmhaltcl -f</td>
<td>This command will halt daemons on all currently running systems</td>
<td>will not re-enable switching on a recovery package if any of the primary, data receiver, or data sender package is in the same recovery group is running or enabled. will not start a primary, data receiver, or data sender package if the recovery package in the same recovery group is running or enabled</td>
</tr>
</tbody>
</table>
Designing a Disaster Tolerant Architecture for use with ContinentalClusters

The ContinentalClusters product operates as a configuration of two MC/ServiceGuard clusters, which can run a package on a cluster and a Recovery Cluster. The key elements providing disaster tolerance in a continental cluster are:

- Mutual Recovery
- MC/ServiceGuard clusters
- Data replication
- Highly available WAN networking
- Data center processes and procedures coordinated between the two cluster sites

You have a great deal of latitude in selecting these elements for your configuration. It is recommended that you record your choices on worksheets which can be reviewed and updated periodically.

Mutual Recovery

For mutual recovery, any cluster in a continental cluster may contain both primary and recovery packages for any recovery group. Recovery groups may be defined, for example, such that cluster A and cluster B contain recovery packages. In this case, `cmrecovercl` could be run on cluster B to recover packages from cluster A, or on cluster A to recover packages from cluster B.

MC/ServiceGuard Clusters

Each MC/ServiceGuard cluster in a continental cluster provides high availability for an application at the local level at that particular site. For optimal performance and to assure adequate capacity on the recovery cluster, it is best to have similar hardware on both clusters. For example, if one cluster contains two V class HP 9000 systems with 1Gb of memory each, it is not a good idea to have a low-end K series HP 9000 with 128 Mb of memory in the other cluster. Each cluster may have as many nodes as are permitted in an ordinary MC/ServiceGuard cluster, and each may...
Building a Continental Cluster

Designing a Disaster Tolerant Architecture for use with Continental Clusters

be running packages that are not configured to fail over between clusters.

NOTE

Remember that when cluster A takes over for cluster B, it must run cluster B's packages as well as any packages that it was already running on its own, unless you choose to stop those packages.

Data Replication

Data replication between the MC/ServiceGuard clusters extends the scope of high availability to the level of the continental cluster. You must select a technology for data replication between the two clusters. There are many possible choices, including:

- Logical replication of databases
- Logical replication of filesystems
- Physical replication of data volumes via software
- Physical replication of disk units via hardware

Table 5-3 is a brief discussion of how a data replication method affects a continental cluster environment. A detailed description of data replication can be found in Chapter 1, in the section titled "Disaster Tolerant Architecture Guidelines." Specific guidelines for configuring the HP SureStore E Disk Array XP Series and the EMC Symmetrix Disk Array for physical data replication in a continental cluster are provided in Chapters 6 and 7. White papers describing specific implementations...
Building a Continental Cluster

Designing a Disaster Tolerant Architecture for use with ContinentalClusters

are also available from http://docs.hp.com/hpux/ha.

Table 5-3

Data Replication and ContinentalClusters

<table>
<thead>
<tr>
<th>Replication Type</th>
<th>How it Works</th>
<th>ContinentalClusters Implication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logical Database Replication</td>
<td>Transactions from the primary application are applied from logs to a copy of the application running on the recovery site. (This is an example only; there are other methods.)</td>
<td>Requirements on CPU and I/O may limit or prevent the Recovery Cluster from running additional applications.</td>
</tr>
<tr>
<td>Logical Filesystem Replication</td>
<td>Writes to the filesystem on the primary cluster are duplicated periodically on the recovery cluster.</td>
<td>CPU issues are the same as for Logical Database Replication. The software may have to be managed as a separate MC/ServiceGuard package.</td>
</tr>
<tr>
<td>Physical Replication of Data Volumes via Software</td>
<td>Disk mirroring via LVM software. Only limited distances are possible (up to 10 km), since mirroring is done on disk links (SCSI or FibreChannel).</td>
<td>Requirements on CPU are less than for logical replication, but there is still some CPU use. Distance limits may make this type of replication inappropriate for ContinentalClusters.</td>
</tr>
<tr>
<td>Physical Replication of Disk Units via Hardware</td>
<td>Replication of the LUNs within a disk array through dedicated hardware links such as EMC SRDF or Continuous Access XP.</td>
<td>Limited CPU requirements, but the requirement of synchronous data replication slows replication, and may impair application performance. Increased network speed and bandwidth can remedy this.</td>
</tr>
</tbody>
</table>

Logical data replication may require the use of packages to handle software processes that copy data from one cluster to another or that apply transactions from logs that are copied from one cluster to another. Some methods of logical data replication may use a logical replication data sender package; others may use a logical replication data receiver package; some may use both. Logical replication data sender and receiver packages are configured as part of the data recovery group, as shown below under “Creating the ContinentalClusters.”
Building a Continental Cluster

Designing a Disaster Tolerant Architecture for use with ContinentalClusters

Configuration.”

Physical Data Replication using Special Control Scripts

Physical data replication generally does not require the use of separate sender or receiver packages, but it does require specialized logic in the package control scripts to handle the transfer of control from the storage units of one cluster to the storage units at the other cluster. Control scripts for packages that use physical data replication with the HP SureStore E Disk Array XP Series with Continuous Access XP should be created using template /opt/cmcluster/toolkit/SGCA/sgcapkg.cntl; control scripts for packages that are using physical data replication with EMC Symmetrix and the SRDF facility should be created using /opt/cmcluster/toolkit/SGSRDF/srdfpkg.cntl. Both of these templates are included with the ContinentalClusters product.

Details on configuring the special ContinentalClusters control scripts are in Chapters 6 and 7. Some additional notes are provided below.

Highly Available Wide Area Networking

Disaster tolerant networking for ContinentalClusters is directly tied to the data replication method. In addition to the reliability of the redundant lines connecting the remote nodes, you also need to consider what bandwidth you need to support the data replication method you have chosen. A continental cluster that handles a high number of write transactions per minute will not only require a highly available network, but also one with a large amount of bandwidth. Details on highly available networking can be found in Chapter 1, in the section titled “Disaster Tolerant Architecture Guidelines.” White papers describing specific implementations are also available from http://docs.hp.com.

Data Center Processes

ContinentalClusters provides the cmrecovercl command that fails over all applications on the primary cluster that are protected by ContinentalClusters. However, application failover also requires well-defined processes for the two sites. These processes and procedures should be written down and made available at both sites.

Some considerations for site management are as follows:

• Who notifies whom for the various events: configuration changes, alerts, alarms?
Building a Continental Cluster

Designing a Disaster Tolerant Architecture for use with ContinentalClusters

• What communication methods should be used? Email? Phone? Beeper? Multiple methods?

• Who has authority to perform what sort of configuration modifications? Can the administrator at one site log in to the nodes on the remote site? If so, what permissions would be set?

• How often is a practice failover done?

• Is there a documented test plan?

• What is the process for tracking changes made to the primary cluster?

ContinentalClusters Worksheets

Planning is an essential effort in creating a robust continental cluster environment. It is recommended that you record the details of your configuration on planning worksheets. These worksheets can be filled in partially before configuration begins, and then completed as you build the continental cluster. Both the site with the cluster and the site with the Recovery Cluster should have a copy of these worksheets to help coordinate initial configuration and subsequent changes. Complete the worksheets in the following sections for each pair of clusters that will be monitored by the ContinentalClusters monitor.
Building a Continental Cluster
Designing a Disaster Tolerant Architecture for use with ContinentalClusters

Data Center Worksheet

The following worksheet will help you describe your specific data center configuration. Fill out the worksheet and keep it for future reference.

=======================================================================
Continental Cluster Name: _____________________________________________
=======================================================================

Primary Data Center Information:

Primary Cluster Name: ____________________________________________
Data Center Name and Location: ___________________________________
Main Contact: ____________________________________________________
Phone Number: ____________________________________________________
Beeper: __________________________________________________________
Email Address: ___________________________________________________
Node Names: ______________________________________________________
Monitor Package Name: __ccmonpkg__________________________________
Monitor Interval: __60 seconds____________________________________

=======================================================================
Recovery Data Center Information:

Recovery Cluster Name: _____________________________________________
Data Center Name and Location: ___________________________________
Main Contact: ____________________________________________________
Phone Number: ____________________________________________________
Beeper: __________________________________________________________
Email Address: ___________________________________________________
Node Names: ______________________________________________________
Monitor Package Name: __ccmonpkg__________________________________
Monitor Interval: __60 seconds____________________________________
Recovery Group Worksheet

The following worksheet will help you organize and record your specific recovery groups. Fill out the worksheet and keep it for future reference.

=======================================================================
Continental Cluster Name: _____________________________________________
=======================================================================

Recovery Group Data:

Recovery Group Name: _____________________________________________
Primary Cluster/Package Name:_____________________________________
Data Sender Cluster/Package Name:_______________________________
Recovery Cluster/Package Name:____________________________________
Data Receiver Cluster/Package Name:_______________________________

Recovery Group Data:

Recovery Group Name: _____________________________________________
Primary Cluster/Package Name:_______________________________
Data Sender Cluster/Package Name:_______________________________
Recovery Cluster/Package Name:____________________________________
Data Receiver Cluster/Package Name:_______________________________

Recovery Group Data:

Recovery Group Name: _____________________________________________
Primary Cluster/Package Name:_______________________________
Data Sender Cluster/Package Name:_______________________________
Recovery Cluster/Package Name:____________________________________
Data Receiver Cluster/Package Name:_______________________________
Building a Continental Cluster

Designing a Disaster Tolerant Architecture for use with Continental Clusters

Cluster Event Worksheet

The following worksheet will help you organize and record the cluster events you wish to track. Fill out a worksheet for each primary or recovery cluster that you wish to monitor. You must monitor each cluster containing a recovery package.

<table>
<thead>
<tr>
<th>Continental Cluster Name:</th>
<th>______________________________________________________</th>
</tr>
</thead>
</table>

Cluster Event Information:

<table>
<thead>
<tr>
<th>Cluster Name</th>
<th>Monitoring Cluster</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>UNREACHABLE:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alert Interval:</td>
</tr>
<tr>
<td>Alarm Interval:</td>
</tr>
<tr>
<td>Notification:</td>
</tr>
<tr>
<td>Notification:</td>
</tr>
<tr>
<td>Notification:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DOWN:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alert Interval:</td>
</tr>
<tr>
<td>Notification:</td>
</tr>
<tr>
<td>Notification:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>UP:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alert Interval:</td>
</tr>
<tr>
<td>Notification:</td>
</tr>
<tr>
<td>Notification:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ERROR::</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alert Interval:</td>
</tr>
<tr>
<td>Notification:</td>
</tr>
<tr>
<td>Notification:</td>
</tr>
<tr>
<td>Notification:</td>
</tr>
</tbody>
</table>
Preparing the Clusters

The steps for configuring the clusters needed by ContinentalClusters are as follows:

- Set up and test data replication between the sites.
- Configure each cluster for MC/ServiceGuard operation.

Setting up and Testing Data Replication

Depending on which data replication method you choose, it can take a week or more to set up and test a data replication method.

In the sample configuration, physical data replication is done through a hardware link between disk arrays. Because this method is hardware based, there is hardware set up and configuration that can take several days. Some logical replication methods, such as transaction processing monitors (TPMs), need application changes that are more easily done during the original application development.

Make sure that the data replication to the recovery site is functional. This would include setting up the ESCON links across the WAN and making sure that data is replicated between the shared disk arrays.

NOTE

If you are using physical data replication on the HP SureStore Disk Array XP Series with Continuous Access XP or on the EMC Symmetrix using EMC SRDF, use the special package control script templates for those products that are installed along with ContinentalClusters software. Refer to Chapters 6 and 7 for detailed instructions on configuring these special control scripts.

If the data replication software is separate from the application itself, then a separate MC/ServiceGuard package should be created for it. Some kinds of logical data replication require that a data receiver package be running on the recovery cluster at all times. If data sender and data receiver packages are used as your choice of data replication method, configure and apply them as described in the next sections before applying the continental cluster configuration.
Building a Continental Cluster
Preparing the Clusters

Table 5-4 shows the types of packages that are needed for each type of data replication.

Table 5-4
ContinentalClusters Data Replication Package Structure

<table>
<thead>
<tr>
<th>Replication Type</th>
<th>Primary Cluster</th>
<th>Recovery Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Primary Application Package</td>
<td>Data Replication Sender Package</td>
</tr>
<tr>
<td>XP Series/CA</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Symmetrix/EMC SRDF</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Oracle Standby Database</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

Configuring a Cluster without Recovery Packages

Use the following steps and the instructions in chapters 4 through 7 of Managing MC/ ServiceGuard (B3936-90045) as guidelines for creating a new cluster or preparing an existing cluster to run in a ContinentalClusters environment:

1. If you are creating a new cluster, install required versions (or later) of HP-UX and MC/ServiceGuard. If you are using an existing cluster, upgrade to the versions of HP-UX and MC/ServiceGuard that are required for ContinentalClusters. See the ContinentalClusters Release Notes for specifics on required version. Coordinate with the recovery site to make sure the same versions and patches are installed at both sites.

2. Set up all cabling, being sure to provide redundant disk storage links and network connections.

3. Configure the disks and filesystems. Set up data replication (logical or physical).

4. Configure the cluster according to the instructions in chapter 5 of Managing MC/ ServiceGuard (B3936-90045). Use the cmapplyconf command to apply the cluster configuration. Then test the cluster.
5. Configure and test each primary package according to the instructions in chapters 6 and 7 of Managing MC/ServiceGuard (B3936-90045). Use the `cmapplyconf` command to apply the package configuration.

6. Be sure that **AUTO_RUN (PKG_SWITCHING_ENABLED)** is set to **NO** in the package ASCII configuration file for any package that is in a recovery group, and therefore might at some time be a candidate for recovery. This is to ensure that the package will not be automatically started if the primary site tries to come up again following a primary site disaster.

   If you change the setting of the **PKG_SWITCHING_ENABLED** parameter to **NO** in the ASCII configuration file for an existing package, you must re-apply the configuration using the `cmapplyconf` command.

   **NOTE** When package switching is disabled, a package does not automatically start at cluster startup time. Therefore, setting **AUTO_RUN (PKG_SWITCHING_ENABLED)** to **NO** means that packages in recovery groups must be started manually on the primary cluster. They also must be manually enabled for local switching, if that is desired, using the `cmmmodpkg -e <packagename>` command.

7. Test local failover of the packages. In our sample case, this would mean enabling package switching for `salespkg` (`cmmmodpkg -e salespkg`) and then testing that `salespkg` fails over from `LANode1` to `LANode2`.

8. If you are using logical data replication, configure and test the data sender package if one is needed.

The primary cluster is shown in Figure 5-4.
Configuring a Cluster with Recovery Packages

Use the following steps and the instructions in chapters 4 through 7 of Managing MC/ServiceGuard (B3936-90045) as guidelines for creating a new Recovery Cluster or preparing an existing cluster to run in a ContinentalClusters environment:

1. Configure all hardware. Make sure the cluster hardware is able to handle the task of running any or all packages it supports in the ContinentalClusters configuration:
   a. If this is a new cluster, make sure the hardware is similar to that of the other cluster. For example, if V class HP 9000 systems comprise one cluster, they should be used to build the other.
   b. If this is an existing cluster, determine whether you need to add disks for data replication. This is needed to ensure that there is enough capacity from system resources to run all packages if applications fail over to the other cluster. If not, either add nodes to the existing cluster, or move less critical packages to another cluster.

2. For new clusters, install minimum required versions of HP-UX and MC/ServiceGuard. For existing clusters, perform a rolling upgrade to the minimum required versions of HP-UX and MC/ServiceGuard if necessary. Coordinate with the other site to make sure the same
versions and patches are installed at both sites. This may include coordinating between HP support personnel if the sites have separate support contracts.

3. Configure logical volumes, using the same names on both the clusters whenever possible. If your cluster uses a physical data replication method and if data replication between the disk arrays at the different data centers has already taken place, `vgimport` and `vgchange` can be used to help configure the logical volumes on the Recovery Cluster.

4. Use `cmgetconf` to capture the other cluster's configuration. Then use `cmquerycl` on this cluster to generate a new ASCII file for the recovery configuration. Modify the node names, volume group names, resource names, and subnets as appropriate so that the two clusters will be consistent. See chapter 5 in Managing MC/ServiceGuard (B3936-90045) for details on cluster configuration.

5. Set up the recovery package(s):
   a. Copy the package files from the other cluster for all mission critical applications to be monitored by ContinentalClusters. In the sample configuration this means copying the ASCII files `salespkg.config` and `custpkg.config`, and the control scripts `salespkg.cntl` and `custpkg.cntl`. (You may want to rename the package configuration files using a naming convention that lets you know that a package is a ContinentalClusters monitored package. For example, you may want to name the sample package `salespkg_bak.config` to indicate that it is the backup or recovery package.)
   
   b. Edit the package configuration files, replacing node names, subnets, and other elements as needed. For all recovery packages, be sure that `AUTO_RUN` (`PKG_SWITCHING_ENABLED` used prior to ServiceGuard 11.12) is set to `NO` in the configuration file. This will ensure that the recovery packages will not start automatically when the recovery cluster forms, but only when the `cmrecovercl` command is issued.

The following elements should be the same in the package configuration for both the primary and recovery packages:

- Package services
- Failfast settings
Building a Continental Cluster

Preparing the Clusters

c. Modify the package control script (salespkg_bak.cntl), checking for anything that may be different between clusters:

- Volume groups (VGs) may be different.
- IP addresses will be different.
- Site-specific customer-defined routines (for example routines that send messages to a local administrator) may be different.
- Control script files must be executable.

NOTE
If you are using physical data replication on the HP SureStore Disk Array XP Series with Continuous Access XP or on the EMC Symmetrix using EMC SRDF, use the special package control script templates for those products that are installed along with ContinentalClusters software.

6. Apply the configuration using cmapplyconf and test the cluster.
7. Test local failover of the packages. In our sample case, this would mean enabling package switching for salespkg_bak (cmmodpkg -e salespkg_bak) and then testing that salespkg_bak fails over from NYnode1 to NYnode2.
8. If you are using logical data replication, configure, apply, and test the data receiver package if one is needed.

The New York cluster is shown in Figure 5-5.

Figure 5-5 Sample Cluster Configuration with Recovery Packages
Building the ContinentalClusters Configuration

If necessary, use the `swinstall` command to install the ContinentalClusters product on all nodes in both clusters. Then create the ContinentalClusters configuration using the following steps (each step is described in detail in the sections that follow):

- Prepare the security files.
- Create the monitor package on each cluster containing a recovery package. Clusters not containing a recovery package may also monitor the other cluster by creating a monitor package on that cluster.
- Edit the ContinentalClusters configuration file on a node of your choice in either cluster.
- Check and apply the ContinentalClusters configuration.
- Start each ContinentalClusters monitor package on its cluster.
- Validate the configuration.
- Document the recovery procedure and distribute the documentation to both sites. Make sure all personnel are familiar with these procedures.
- Test recovery procedures.

Preparing Security Files

Configuring a continental cluster requires root access to all the nodes in both clusters in the configuration. Before doing the ContinentalClusters configuration, edit the `~/.rhosts` file on all the nodes of both clusters to include entries that will allow access by the node on which you will run the `cmapplyconcl` command. Here is a sample entry in the `~/.rhosts` file that allows the root user on system3 to run the `cmapplyconcl` command on the node where the `~/.rhosts` file contains the entry:

```
system3.westcoast.myco.com root
```

After the `cmapplyconcl` command has been run successfully, you can remove this entry from the `~/.rhosts` file if you wish. Remember, however, that the entry must be present in the `~/.rhosts` file when you...
Building a Continental Cluster

Building the Continental Clusters Configuration

use `cmapplyconcl` at a later time.

---

**NOTE**

The `cmclnodelist` file does not provide the required type of access for the `cmapplyconcl` command.

---

You must also create the `/etc/opt/cmom/cmomhosts` file on all nodes. This file allows nodes that are running monitor packages to obtain information from other nodes about the health of each cluster. The file must contain entries that allow access to all nodes in the continental cluster by the nodes where monitors are running.

You define the order of security checking by creating entries of the following types:

**order deny,allow** If `deny` is first, the deny list is checked first to see if the node is there, then the allow list is checked.

**deny from** lists all the nodes that are denied access. Permissible entries are:

- **all** All hosts are denied access.
- **domain** Hosts whose names match, or end in, this string are allowed access, e.g. `hp.com`.
- **hostname** The named host (for example, `kitcat.myco.com`) is denied access.
- **IP address** Either a full IP address, or a partial IP address of 1 to 3 bytes for subnet restriction is allowed.
- **network/netmask** This pair of addresses allows more precise restriction of hosts, (e.g. `10.163.121.23/225.225.0.0`).
- **network/nnn CIDR** This specification is like the network/netmask specification, except the netmask consists of `nnn` high-order 1 bits. “CIDR” stands for classless interdomain routing, a type of routing supported by the Border Gateway protocol (BGP).
allow from lists all the nodes that are allowed access. Permissible entries are:

- **all**: All hosts are allowed access.
- **domain**: Hosts whose names match, or end in, this string are allowed access, e.g. hp.com.
- **hostname**: The named host (for example, kitcat.myco.com) is allowed access.
- **IP address**: Either a full IP address, or a partial IP address of 1 to 3 bytes for subnet inclusion is allowed.
- **network/netmask**: This pair of addresses allows more precise inclusion of hosts, (e.g. 10.163.121.23/225.225.0.0).
- **network/nnn CIDR**: This specification is like the network/netmask specification, except the netmask consists of nnn high-order 1 bits. “CIDR” stands for classless interdomain routing, a type of routing supported by the Border Gateway protocol (BGP).

The most typical entry is `hostname`. The following entries are from a typical `/etc/opt/cmom/cmomhosts` file:

```plaintext
order allow,deny
allow from lanode1.myco.com
allow from lanode2.myco.com
allow from nynode1.myco.com
allow from nynode2.myco.com
allow from 10.177.242.12
```

If the file is installed on all nodes in the continental cluster, these entries will allow the monitors running on lanode1, lanode2, nynode1, and nynode2 to obtain information about the health of all nodes in the configuration.
Creating the Monitor Package

The ContinentalClusters monitoring software is configured as an MC/ServiceGuard package so that it remains highly available. The following steps should be carried out on the recovery cluster and repeated on the primary cluster if you want to monitor the recovery site from the primary site:

1. On the node where you are doing the configuration, create a directory for the monitor package:
   
   ```
   # mkdir /etc/cmcluster/ccmonpkg
   ```

2. Copy the template files from the `/opt/cmconcl/scripts` directory to the `/etc/cmcluster/ccmonpkg` directory:
   
   ```
   # cp /opt/cmconcl/scripts/* /etc/cmcluster/ccmonpkg
   ```

   - `ccmonpkg.config` is the ASCII package configuration file template for the ContinentalClusters monitoring application.
   - `ccmonpkg.cntl` is the control script file for the ContinentalClusters monitoring application.

   **NOTE** You do not need to edit `ccmonpkg.cntl`.

3. Edit the package configuration file (suggested name of `/etc/cmcluster/ccmonpkg/ccmonpkg.config`) to match the cluster configuration:
   
   a. Add the names of all nodes in the cluster on which the monitor may run.

   b. `AUTO_RUN (PKG_SWITCHING_ENABLED used prior to ServiceGuard 11.12) should be set to YES so that the monitor package will fail over between local nodes. (Note, however, that for all primary and recovery packages, AUTO_RUN is always set to NO.)`

4. Use the `cmcheckconf` command to validate the package:
   
   ```
   # cmcheckconf -P ccmonpkg.config
   ```

5. Use the `cmapplyconf` command to add the package to the MC/ServiceGuard configuration:
   
   ```
   # cmapplyconf -P ccmonpkg.config
   ```
6. Copy the package control script ccmonpkg.cntl to the monitor package directory (default name /etc/cmcluster/ccmonpkg) on all the other nodes in the cluster. Make sure this file is executable.

The following sample package configuration file (comments have been left out) shows a typical package configuration for a ContinentalClusters monitor package:

```bash
#PACKAGE_NAME ccmonpkg
FAILOVER_POLICY CONFIGURED_NODE
FAILBACK_POLICY MANUAL
NODE_NAME LAnode1
NODE_NAME LAnode2
RUN_SCRIPT /etc/cmcluster/ccmonpkg/ccmonpkg.cntl
RUN_SCRIPT_TIMEOUT NO_TIMEOUT
HALT_SCRIPT /etc/cmcluster/ccmonpkg/ccmonpkg.cntl
HALT_SCRIPT_TIMEOUT NO_TIMEOUT
SERVICE_NAME ccmonpkg.srv
SERVICE_FAIL_FAST_INCLUDED NO
SERVICE_HALT_TIMEOUT 300
AUTO_RUN YES
NET_SWITCHING_ENABLED YES
NODE_FAIL_FAST_INCLUDED NO
```

**Editing the ContinentalClusters Configuration File**

First, on one cluster, generate an ASCII configuration template file using the cmqueryconcl command. The recommended name and location for this file is /etc/cmcluster/cmconcl.config. (You can choose a different name if you wish.) Example:

```bash
# cd /etc/cmcluster
# cmqueryconcl -C cmconcl.config
```

This file has three editable sections:

- Cluster information
- Recovery groups
- Monitoring definitions

Customize each section according to your needs. The following are some guidelines for editing each section.
Building a Continental Cluster

Building the ContinentalClusters Configuration

Editing Section 1—Cluster Information

Enter cluster-level information as follows in this section of the file:

1. Enter a name for the continental cluster on the line that contains the 
   CONTINENTAL_CLUSTER_NAME keyword. This can be any name you 
   choose, but it cannot be changed after the configuration is applied. To 
   change the name, you must first delete the existing configuration as 
   described in "Renaming a Continental Cluster" on page 202.

2. Enter the name of the first cluster after the first CLUSTER_NAME 
   keyword followed by the names of all the nodes within the first 
   cluster. Use a separate NODE_NAME keyword and HP-UX host name for 
   each node.

3. Enter the domain name of the cluster’s nodes following the 
   DOMAIN_NAME keyword.

4. Optionally, enter the name of the monitor package on the first cluster 
   after the MONITOR_PACKAGE_NAME keyword and the interval at which 
   monitoring by this package will take place (minutes and/or seconds) 
   following the MONITOR_INTERVAL keyword.

The monitor interval defines how long it can take for 
ContinentalClusters to detect that a cluster is in a certain state. The 
default interval is 60 seconds, but the optimal setting depends on 
your system’s performance. Setting this interval too low can result in 
the monitor’s falsely reporting an Unreachable or Error state. If you 
observe this during testing, use a larger value.

It is suggested that you use the name “ccmonpkg” for all 
ContinentalClusters monitors. Create this package on each cluster 
containing a recovery package. If you do not wish to monitor a cluster, 
not containing a recovery package, you must delete or comment out 
the MONITOR_PACKAGE_NAME line and the MONITOR_INTERVAL line. For 
mutual recovery, create the monitor package on both the first and 
second clusters.

NOTE Monitoring of a cluster not containing recovery packages is optional. For 
extample, you might set up monitoring of such a cluster so you can check 
the status of the data replication technology being used.

5. Repeat steps 2 through 4 for the alternate cluster.
The monitor package is sensitive to system time and date. If you change the system time or date either backwards or forwards on the node where the monitor is running, notifications of alerts and alarms may be sent at incorrect times.

A printout of Section 1 of the ContinentalClusters ASCII configuration file follows.

```plaintext
CONTINENTAL CLUSTER CONFIGURATION FILE

This file contains ContinentalClusters configuration data. The file is divided into three sections, as follows:

1. Cluster Information
2. Recovery Groups
3. Events, Alerts, Alarms, and Notifications

For complete details about how to set the parameters in this file, consult the cmqueryconcl(1m) manpage or your manual.

Section 1. Cluster Information

This section contains the name of the continental cluster followed by the names of member clusters and all their nodes. The continental cluster name can be any string you choose, up to 40 characters in length. Each member cluster name must be the same as it appears in the MC/ServiceGuard cluster configuration ASCII file for that cluster. In addition to the cluster name, include a domain name for the nodes in the cluster. Node names must be the same as those that appear in the cluster configuration ASCII file.

In the space below, enter the continental cluster name, then enter a cluster name and domain for each member cluster, and the names of all the nodes in that cluster. Following the node names, enter the name of a monitor package that will run the continental cluster monitoring software on that cluster. It is strongly recommended that you use the same name for the monitoring package on all clusters; "ccmonpkg" is suggested. Monitoring of the recovery cluster by the primary cluster is optional. If you do not wish to monitor the recovery cluster, you must delete or comment out the MONITOR_PACKAGE_NAME and MONITOR_INTERVAL lines that follow the name of the primary cluster.

After the monitor package name, enter a monitor interval,
Building a Continental Cluster

Building the Continental Clusters Configuration

### specifying a number of minutes and/or seconds. The default is 60 seconds, the minimum is 30 seconds, and the maximum is 5 minutes. ###
### Example: ###
### CONTINENTAL_CLUSTER_NAME ccluster1 ###
### CLUSTER_NAME westcoast ###
### CLUSTER_DOMAIN westnet.myco.com ###
### NODE_NAME system1 ###
### NODE_NAME system2 ###
### MONITOR_PACKAGE_NAME ccmonpkg ###
### MONITOR_INTERVAL 1 MINUTE 30 SECONDS ###
### CLUSTER_NAME eastcoast ###
### CLUSTER_DOMAIN eastnet.myco.com ###
### NODE_NAME system3 ###
### NODE_NAME system4 ###
### MONITOR_PACKAGE_NAME ccmonpkg ###
### MONITOR_INTERVAL 1 MINUTE 30 SECONDS ###

CONTINENTAL_CLUSTER_NAME ccluster1

CLUSTER_NAME

   CLUSTER_DOMAIN
   NODE_NAME
   NODE_NAME
   MONITOR_PACKAGE_NAME ccmonpkg
   MONITOR_INTERVAL 60 SECONDS

CLUSTER_NAME

   CLUSTER_DOMAIN
   NODE_NAME
   NODE_NAME
   MONITOR_PACKAGE_NAME ccmonpkg
   MONITOR_INTERVAL 60 SECONDS

Editing Section 2—Recovery Groups

In this section of the file, you define recovery groups, which are sets of ServiceGuard packages that are ready to recover applications in case of cluster failure. You create a separate recovery group for each package that will be started on a cluster when the cmrecovercl(1m) command is issued on that cluster.

Examples of recovery groups are shown graphically in Figure 5-6 and Figure 5-7.
Figure 5-6  Sample ContinentalClusters Recovery Groups

**Recovery Group for Sales Application:**

- **RECOVERY_GROUP_NAME:** Sales
- **PRIMARY_PACKAGE:** LAcluster/salespkg
- **RECOVERY_PACKAGE:** NYcluster/salespkg_bak

**Recovery Group for Customer Application:**

- **RECOVERY_GROUP_NAME:** Customer
- **PRIMARY_PACKAGE:** LAcluster/custpkg
- **RECOVERY_PACKAGE:** NYcluster/custpkg_bak

**Los Angeles Cluster**

- LANode1
  - salespkg.config
  - salespkg.cntl

- LANode2
  - custpkg.config
  - custpkg.cntl

**New York Cluster**

- NYnode1
  - salespkg_bak.config
  - salespkg_bak.cntl
  - custpkg_bak.config
  - custpkg_bak.cntl

- NYnode2
  - custpkg_bak.config
  - custpkg_bak.cntl

**WAN**
Building a Continental Cluster
Building the ContinentalClusters Configuration

Figure 5-7 Sample Bi-directional Recovery Groups

Enter data in Section 2 as follows:

1. Enter a name for the recovery group following the `RECOVERY_GROUP_NAME` keyword. This can be any name you choose.

2. After the `PRIMARY_PACKAGE` keyword, enter a primary package.
Building the ContinentalClusters Configuration

3. Optionally, enter a data sender package definition consisting of the cluster name, a slash (/), and the data sender package name after the DATA_SENDER_PACKAGE keyword. This is only necessary if you are using a logical data replication method that requires a data sender package.

4. After the RECOVERY_PACKAGE keyword, enter a recovery package definition consisting of the cluster name followed by a slash (/) followed by the package name. Example:

   RECOVERY_PACKAGE NYcluster/custpkg_bak

5. Optionally, enter a data receiver package definition consisting of the cluster name, a slash (/), and the data receiver package name after the DATA_RECEIVER_PACKAGE keyword. This is only necessary if you are using a logical data replication method that requires a data receiver package.

6. Repeat these steps for each package that will be recovered. Each package must be configured in a separate recovery group.

A printout of Section 2 of the ContinentalClusters ASCII configuration file follows.

******************************************************************************
## Section 2. Recovery Groups
##
## This section defines recovery groups--sets of ServiceGuard packages that are ready to recover applications in case of cluster failure. Recovery groups allow one cluster in the continental cluster configuration to back up another member cluster’s packages. You create a separate recovery group for each ServiceGuard package that will be started on the recovery cluster when the cmrecovercl(1m) command is issued.

## A recovery group consists of a primary package running on one cluster, and a recovery package that is ready to run on a different cluster. In some cases, a data receiver package runs on the same cluster as the recovery package, and in some cases, a data sender package runs on the same cluster as the primary package.

## During normal operation, the primary package is running an application program on the primary cluster, and the recovery package, which is configured to run the same application, is idle on the recovery cluster. If the primary package performs

Chapter 5  169
Building a Continental Cluster

Building the ContinentalClusters Configuration

---

### disk I/O, the data that is written to disk is replicated
### and made available for possible use on the recovery cluster.
### For some data replication techniques, this involves the use of
### a data receiver package running on the recovery cluster.
### In the event of a major failure on the primary cluster, the
### user issues the cmrecovercl(1m) command to halt any data
### receiver packages and start up all the recovery packages
### that exist on the recovery cluster.
### Enter the name of each package recovery group together with
### the fully qualified names of the primary and recovery
### packages. If appropriate, enter the fully qualified name
### of a data receiver package. Note that the data receiver
### package must be on the same cluster as the recovery package.
### The primary package name includes the primary cluster name
### followed by a slash ("/"), followed by the package name on
### the primary cluster. The recovery package includes
### the recovery cluster name, followed by a slash ("/"),
### followed by the package name on the recovery cluster.
### The data receiver package name includes the recovery cluster
### name, followed by a slash ("/"), followed by the name of
### the data receiver package on the recovery cluster.
### Up to 29 recovery groups can be entered.
### Example:
### RECOVERY_GROUP_NAME       nfsgroup
### PRIMARY_PACKAGE            westcoast/nfspkg
### DATA_SENDER_PACKAGE        westcoast/nfssenderpkg
### RECOVERY_PACKAGE           eastcoast/nfsbackuppkg
### DATA_RECEIVER_PACKAGE      eastcoast/nfsreceiverpkg
### RECOVERY_GROUP_NAME       hpgroup
### PRIMARY_PACKAGE            westcoast/hppkg
### DATA_SENDER_PACKAGE        westcoast/hpsenderpkg
### RECOVERY_PACKAGE           eastcoast/hpbackuppkg
### DATA_RECEIVER_PACKAGE      eastcoast/hpreceiverpkg

---

Editing Section 3—Monitoring Definitions

Finally, you enter monitoring definitions, which define cluster events and
set times at which alert and alarm notifications are to be sent out. Define
notifications for all cluster events—Unreachable, Down, Up, and Error.
Although it is impossible to make specific recommendations for every
Building a Continental Cluster
Building the ContinentalClusters Configuration
ContinentalClusters environment, here are a few general guidelines
about notifications.
1. Specify the cluster event by using the CLUSTER_EVENT keyword
followed by the name of the cluster, a slash (“/”) and the name of the
status—Unreachable, Down, Up, or Error. Example:
CLUSTER_EVENT LAcluster/UNREACHABLE
2. Define a CLUSTER_ALERT at appropriate times following the
appearance of the event. Specify the elapsed time and include a
NOTIFICATION message that provides useful information about the
event. You can create as many alerts as needed, and you can send as
many notifications as you wish to different destinations (see the
comments in the file excerpt below for a list of destination types).
Note that the message text in the notification must be on a separate
line in the file.
3. If the event is for a cluster in an Unreachable condition, define a
CLUSTER_ALARM at appropriate times. Specify the elapsed time since
the appearance of the event (greater than the time used for the last
CLUSTER_ALERT), and include a NOTIFICATION message that
indicates what action should be taken. You can create as many
alarms as needed, and you can send as many notifications as you wish
to different destinations (see the comments in the file excerpt below
for a list of destination types).
4. If you are using a monitor on a cluster containing no recovery
packages define alerts for the monitoring of Up, Down, Unreachable,
and Error states on the recovery cluster. It is not necessary to define
alarms.
A printout of Section 3 of the ContinentalClusters ASCII configuration
file follows.
###############################################################################
####
####
####
Section 3. Monitoring Definitions
####
####
####
####
This section of the file contains monitoring definitions.
####
####
Well planned monitoring definitions will help in making the
####
####
decision whether or not to issue the cmrecovercl(1m) command.
####
####
Each monitoring definition specifies a cluster event along with
####
####
the messages that should be sent to system administrators
####
####
or other IT staff. All messages are appended to the default log
####
####
/etc/cmcluster/cmconcl/eventlog as well as being sent to the
####
####
destination you specify below.
####
####
####
####
A cluster event takes place when a monitor that is located on
####

Chapter 5

171


Building a Continental Cluster
Building the Continental Clusters Configuration

One cluster detects a significant change in the condition of another cluster. The monitored cluster conditions are:

- **UNREACHABLE** - the cluster is unreachable. This will occur when the communication link to the cluster has gone down, as in a WAN failure, or when all nodes in the cluster have failed.

- **DOWN** - the cluster is down but nodes are responding. This will occur when the cluster is halted, but some or all of the member nodes are booted and communicating with the monitoring cluster.

- **UP** - the cluster is up.

- **ERROR** - there is a mismatch of cluster versions or a security error.

A change from one of these conditions to another one is a cluster event. You can define alert or alarm states based on the length of time since the cluster event was observed. Some events are noteworthy at the time they occur, and some are noteworthy when they persist over time. Setting the elapsed time to zero results in a message being sent as soon as the event takes place. Setting the elapsed time to 5 minutes results in a message being sent when the condition has persisted for 5 minutes.

An alert is intended as informational only. Alerts may be sent for any type of cluster condition. For an alert, a notification is sent to a system administrator or other destination. Alerts are not intended to indicate the need for recovery. The cmrecovercl(1m) command is disabled.

An alarm is an indication that a condition exists that may require recovery. For an alarm, a notification is sent, and in addition, the cmrecovercl(1m) command is enabled for immediate execution, allowing the administrator to carry out cluster recovery. An alarm can only be defined for an UNREACHABLE or DOWN condition in the monitored cluster.

A notification defines a message that is appended to the log file /etc/cmcluster/cmconcl/eventlog and sent to other specified destinations, including email addresses, SNMP traps, the system console, or the syslog file. The message string in a notification is entered in double quotes on a separate line; it can be no more than 170 characters long. Enter notifications in one of the following forms:

```
NOTIFICATION CONSOLE <message>
Message written to the console.
```

```
NOTIFICATION EMAIL <address> <message>
Message emailed to a fully...```
Chapter 5

Building the Continental Clusters Configuration

### qualified email address. ###

### NOTIFICATION OPC <level> ###

#### <message> ####

The message is sent to OpenView IT/Operations.

The value of <level> may be 8 (normal), 16 (warning), 64 (minor), 128 (major), 32 (critical).

### NOTIFICATION SNMP <level> ###

#### <message> ####

The message is sent as an SNMP trap.

The value of <level> may be 1 (normal), 2 (warning), 3 (minor), 4 (major), 5 (critical).

### NOTIFICATION SYSLOG ###

#### <message> ####

A notice of the event is appended to the syslog file.

### NOTIFICATION TCP <nodename>:<portnumber> ###

#### <message> ####

Message is sent to a TCP port on the specified node.

### NOTIFICATION TEXTLOG <pathname> ###

#### <message> ####

A notice of the event is written to a user-specified log file. <pathname> must be a full path for the user-specified file.

### NOTIFICATION UDP <nodename>:<portnumber> ###

#### <message> ####

Message is sent to a UDP port on the specified node.

For the cluster event, enter a cluster name followed by a slash (/) and a cluster condition (UP, DOWN, UNREACHABLE, ERROR) that may be detected by a monitor program.

Each cluster event must be paired with a monitoring cluster. Include the name of the cluster on which the monitoring will take place. Events can be monitored from either the primary cluster or the recovery cluster.

Alerts, alarms, and notifications have the following syntax.

### CLUSTER_ALERT <min> MINUTES <sec> SECONDS ###

Delay before the software issues an alert notification about the cluster event.

### CLUSTER_ALARM <min> MINUTES <sec> SECONDS ###

Delay before the software issues
Building a Continental Cluster

Building the Continental Clusters Configuration

### Notification <type> <message>

A string value which is sent from the monitoring cluster for a given event to a specified destination. The <message>, which can be no more than 170 characters, is also appended to the file on the monitoring node in the cluster where the event was detected.

Example:

```
CLUSTER_EVENT westcoast/UNREACHABLE

CLUSTER_ALERT 5 MINUTES

NOTIFICATION EMAIL admin@primary.site
"westcoast unreachable for 5 min. Call secondary site."

CLUSTER_ALERT 10 MINUTES

NOTIFICATION EMAIL admin@primary.site
"westcoast unreachable for 10 min. Call secondary site."

CLUSTER_ALERT 15 MINUTES

NOTIFICATION EMAIL admin@primary.site
"westcoast unreachable for 15 min. Takeover advised."

CLUSTER_EVENT westcoast/UP

CLUSTER_EVENT westcoast/DOWN

CLUSTER_EVENT westcoast/ERROR
```

```
Building the ContinentalClusters Configuration

Selecting Notification Intervals

The monitor interval determines the amount of time between distinct attempts by the monitor to obtain the status of a cluster. The intervals associated with notifications need to be chosen to work in combination with the monitor interval to give a realistic picture of cluster events.

Some combinations are not useful. For example, notification intervals that are smaller than the monitor interval do not make sense, and should be avoided. In the following example, the cluster event will always result in two alerts followed by an alarm. No change of state could possibly be detected at the one-minute, two-minute and three-minute intervals, because the monitor does not check for changes.
Building a Continental Cluster

Building the ContinentalClusters Configuration

until the monitor interval (5 minutes) has been reached.

MONITOR_PACKAGE_NAME ccmonpkg
MONITOR_INTERVAL 5 MINUTES
...
CLUSTER_EVENT LACluster/UNREACHABLE
CLUSTER_ALERT 1 MINUTE
NOTIFICATION CONSOLE
"1 Minute Alert: LACluster Unreachable"
CLUSTER_ALERT 2 MINUTES
NOTIFICATION CONSOLE
"2 Minute Alert: LACluster Still Unreachable"
CLUSTER_ALARM 3 MINUTES
NOTIFICATION CONSOLE
"ALARM: LACluster Unreachable after 3 Minutes: Recovery Enabled"

The following sequence could provide meaningful notifications, since a change of state is possible between notification intervals:

MONITOR_PACKAGE_NAME ccmonpkg
MONITOR_INTERVAL 1 MINUTE
...
CLUSTER_EVENT LACluster/UNREACHABLE
CLUSTER_ALERT 3 MINUTES
NOTIFICATION CONSOLE
"3 Minute Alert: LACluster Unreachable"
CLUSTER_ALERT 5 MINUTES
NOTIFICATION CONSOLE
"5 Minute Alert: LACluster Still Unreachable"
CLUSTER_ALARM 10 MINUTES
NOTIFICATION CONSOLE
"ALARM: LACluster Unreachable after 10 Minutes: Recovery Enabled"

A rule of thumb is that the notification intervals should be multiples of the monitor interval.

Checking and Applying the ContinentalClusters Configuration

After editing the configuration file on the primary cluster, halt any monitor packages that are running, then use the following steps to apply the configuration to all nodes in the continental cluster.
Building the ContinentalClusters Configuration

1. Use the following command to verify the content of the file:

   # cmcheckconcl -v -C cmconcl.config

   This command will verify that all parameters are within range, all fields are filled out, and the entries (such as NODE_NAME) are valid.

2. Use the following command to distribute the ContinentalClusters configuration information to all nodes in the continental cluster:

   # cmapplyconcl -v -C cmconcl.config

   Configuration data is copied to all nodes and in both clusters. This data includes a set of managed object files that are copied to the /var/adm/cmconcl/instances directory on every node in both clusters. All nodes must be booted when the command is issued, although the MC/ServiceGuard cluster may or may not be running.

3. Be sure to make a backup copy of the configuration ascii file after it is applied.

   **NOTE**

   Remember that the /.rhosts file on each node must contain an entry that allows write access. See “Preparing Security Files” on page 159.

   If any problems occur during the execution of cmapplyconcl, you can repeat the command as often as necessary. Issuing the command will delete the existing ContinentalClusters configuration and apply the new one.

   When configuration is finished, your systems should have sets of files similar to those shown in Figure 5-8.
Building a Continental Cluster
Building the ContinentalClusters Configuration

Figure 5-8 ContinentalClusters Configuration Files

Starting the ContinentalClusters Monitor Package

Starting the monitoring package enables all ContinentalClusters functionality. Before you do this, ensure that the primary packages you wish to protect are running normally and that data sender and receiver.
Building the ContinentalClusters Configuration

packages, if they are being used for logical data replication, are working correctly.

If you are using physical data replication, make sure that it is operational.

On each monitoring cluster use the following command to start the monitor package:

```sh
# cmmodpkg -e ccmontpkg
```

Validating the Configuration

The following table shows the status of ContinentalClusters packages when each cluster is running normally and no recovery has taken place.

<table>
<thead>
<tr>
<th>Data Replication Method</th>
<th>Primary Cluster</th>
<th>Recovery Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Data Package</td>
<td>Optional Monitor Package</td>
</tr>
<tr>
<td>Physical—Symmetrix</td>
<td>Running</td>
<td>Not used</td>
</tr>
<tr>
<td>Physical—XP Series</td>
<td>Running</td>
<td>Not used</td>
</tr>
<tr>
<td>Logical—Oracle Standby Database</td>
<td>Running</td>
<td>Not used</td>
</tr>
</tbody>
</table>

Use the following steps to make sure the components are functioning correctly:

1. Use the following command to make sure all daemons are running:

```sh
# ps -ef | grep cmcl
```

Two important ContinentalClusters daemons are `cmclsendryd` and `cmclrmond`.

2. Check the cluster configuration on each cluster using the `cmviewcl -v` command.
Building a Continental Cluster

**Building the ContinentalClusters Configuration**

a. Ensure that each primary package is running correctly.

b. Ensure that data sender packages (if any are used for logical data replication) are running correctly.

c. Ensure that data receiver packages (if any are used for logical data replication) are running correctly.

d. Ensure that the continental cluster monitor package is running correctly on each monitoring cluster.

3. On all nodes, use the `tail -f /adm/syslog/syslog.log` command to check the end of the SYSLOG file for errors.

4. On nodes where packages are running, check all package log files for errors, including application packages and the monitor package.

5. Use the following command to verify the correct operation of the ContinentalClusters daemon:

   ```
   # /opt/cmom/tools/bin/cmreadlog -f /var/adm/cmconcl/sentryd.log
   ```

6. Make sure the ContinentalClusters monitor packages (default name ccmonpkg) on each cluster fails over properly if a node fails.

7. Change each cluster’s state to test that the monitor running on the monitoring cluster will detect the change in status and send notification.

8. View the status of the ContinentalCluster primary and recovery clusters, including configured event data:

   ```
   # cmviewconcl -v
   ```

---

**CAUTION**

You should never issue the `cmrunpkg` command for a recovery package when ContinentalClusters is enabled, because there is no guaranteed way of preventing a package that is running on one cluster from running on the other cluster if the package is started using this command. The potential for data corruption is great.

---

Chapters 6 and 7 contain additional suggestions on testing the data replication and package configuration.

**Documenting the Recovery Procedure**

Once everything is configured and the ContinentalClusters monitor is
Building a Continental Cluster

Building the ContinentalClusters Configuration

running, you must define your recovery procedure and train the administrators and operators at both sites. The checklist in Figure 5-9 is an example of how you might document the recovery procedure.

Figure 5-9  Recovery Checklist

☐ Identify the level of alert that the monitoring site received.

☐ Cluster Alert
☐ Cluster Alarm

☐ Contact the monitored site by phone or beeper to rule out the following:

☐ WAN networking failure, primary cluster and packages are still fine.

☐ Cluster and/or package have come back up but UP notification not yet received by recovery site.

☐ Get authorization from the monitored site using one of the following:

☐ Authorized person contacted:

☐ Director 1
☐ Admin 1

☐ Authorization received:

☐ Human-to-human voice authorization
☐ Voice mail

☐ Notify the monitored site of successful recovery using one of the following:

☐ Authorized person contacted:

☐ Director 1
☐ Admin 1

☐ Confirmation received:

☐ Human-to-human voice confirmation
☐ Voice mail

Reviewing the Recovery Procedure

Using the checklist described in the previous section, step through the recovery procedure to make sure that all necessary steps are included. If possible, create simulated failures to test the alert and alarm scenarios coded in the ContinentalClusters configuration file.
Testing the Continental Cluster

This section presents some test procedures and scenarios. Some scenarios presume certain configurations that may not apply to all environments. Additionally, these tests do not eliminate the need to perform standard MC/ServiceGuard testing for each cluster individually.

NOTE
Data and system corruption can occur as a result of testing. System and data backups should always be done prior to testing.

Testing Individual Packages

Use procedures like the following to test individual packages:

1. Use the `cmhaltpkg` command to shut down the package on the primary cluster that corresponds to the package you want to test on the recovery cluster.

2. Do not switch any users to the recovery cluster. The application must be inaccessible to users during this test.

3. Start up the package to be tested on the recovery cluster using the `cmrunpkg` command.

4. Access the application manually using a mechanism that tests network connectivity.

5. Perform read-only actions to verify that the application is running appropriately.

6. Shut down the application on the recovery cluster using the `cmhaltpkg` command.

7. If using physical data replication, do not resync from the recovery cluster to the primary cluster. Instead, manually issue a command that will overwrite any changes on the recovery disk array that may inadvertently have been made.

8. Start the package up on the primary cluster and allow the users to connect to the application.
Testing ContinentalClusters Operations

Use the following procedures to exercise typical ContinentalClusters behaviors.

1. Halt both clusters, then restart both clusters. The monitor packages on both clusters should start automatically. The ContinentalClusters packages (primary, data sender, data receiver, and recovery) should not start automatically. Any other packages may or may not start automatically, subject to their configuration.

   **NOTE**
   If an UP status is configured for a cluster, then an appropriate notification (email, SNMP, etc.) should be received at the configured time interval from the node running the monitor package on the other cluster. Due to delays in email or SNMP, the notifications may arrive later than expected.

   In addition to alerts/alarms sent using the mechanisms defined in the ContinentalClusters configuration file, they are also recorded in the file `/etc/cmcluster/cmconcl/eventlog` on the system reporting the event.

2. While the monitor package is running on a monitoring cluster, halt the monitored cluster (`cmhaltcl -f`). An appropriate notification (email, SNMP, etc.) should be received at the configured time interval from the node running the monitor package. Run `cmrecovercl`. The command should fail. Additional notifications should be received at the configured time intervals. After the alarm notification is received, run `cmrecovercl`. Any data receiver packages on the monitoring cluster should halt and the recovery package(s) should start with package switching enabled. Halt the recovery packages.

3. Test 2 should be rerun under a variety of conditions (and multiple conditions) such as the following:
   - Rebooting and powering off systems one at a time
   - Rebooting and powering off all systems at the same time
   - Running the monitor package on each node in each cluster
   - Disconnecting the WAN connection between the clusters
   - Disconnecting the ESCON connection between the clusters
Building a Continental Cluster

Testing the Continental Cluster

Powering off the disk array at the primary site
Powering off the disk array at the recovery site

- Testing `cmrecovercl -f` as well as `cmrecovercl`

Depending on the condition, the primary packages should be running to test real life failures and recovery procedures.

4. After each scenario in tests 2-4, restore both clusters to their production state, restart the primary package(s) (as well as any data sender and data receiver packages) and note any issues, time delays, etc.

5. Halt the monitor package on one cluster. Halt the other cluster. No notifications are generated that the other cluster has failed. What mechanism is available to the organization to monitor the monitor?

6. Halt the packages on one cluster, but do not halt the cluster. No notifications are generated that the packages on that cluster have failed. What mechanism is available to the organization to monitor package status?

---

**NOTE**

ContinentalClusters monitors cluster status, but not package status.

---

7. View the status of the continental cluster.

    `# cmviewconcl`
Switching to the Recovery Packages in Case of Disaster

Once the clusters are configured and tested, packages will be able to fail over to an alternate node in another data center and still have access to the data they need to function. The primary steps for failing over a package are:

1. Receive notification that a monitored cluster is unavailable.
2. Verify that it is necessary and safe to start the recovery packages.
3. Use the recovery command to stop data replication and start recovery packages.
4. View the status of the continental cluster.

It is crucial that you have a well-defined recovery process, and that all members at both sites are educated on how to use this process.

Receiving Notification

Once the monitor is started, as described in “Starting the Continental Clusters Monitor Package” on page 178, the monitor will send notifications as configured. You may get one of the following types of notification as configured in cmclconf.ascii:

- **CLUSTER_ALERT** is a change in the status of cluster. Recovery via the cmrecovercl command is not enabled by default. This should be treated as information that the cluster either may be developing a problem or may be recovering from a problem.

- **CLUSTER_ALARM** is a change in the status of a cluster that indicates that the cluster has been unavailable for an unacceptable amount of time. Recovery via the cmrecovercl command is enabled.

The issuing of notifications takes place at the timing intervals specified for each cluster event. However, it sometimes may appear that an alert or alarm takes longer than configured. Keep in mind that if several changes of cluster state (for example, Down to Error to Unreachable to Down) take place in a smaller time than the configured interval for an alert or alarm, the timer is reset to 0 after each change of state; thus, the
Building a Continental Cluster

Switching to the Recovery Packages in Case of Disaster

time to the alert or alarm will be the configured interval plus the time used by all the earlier state changes.

NOTE

The cmrecovercl command is fully enabled only after a CLUSTER_ALARM is issued; however, the command may be used with the -f option when a CLUSTER_ALERT has been issued.

Verifying that Recovery is Needed

It is important to follow the established protocol for coordinating with the remote site to determine whether moving the package is required. This includes initiating person-to-person communication between sites. For example, it may be possible that the WAN network failed, causing the cluster alarm.

Some network failures, such as those that prevent clients from using the application, may require recovery. Other network failures, such as those that only prevent the two clusters from communicating, may not require recovery. Following an established protocol for communicating with the remote site would verify this. See Figure 5-9 on page 181 for an example of a recovery checklist.

Using the Recovery Command to Switch All Packages

Once you have received an appropriate notification and have coordinated between the sites (see “Documenting the Recovery Procedure” on page 180 for a sample worksheet,) and have determined that moving the package is necessary, use the cmrecovercl command to start the failover process:

```sh
# cmrecovercl
```

If you have not received a notification defined in a CLUSTER_ALARM statement in the configuration file, but you have received a CLUSTER_ALERT and the remote site has confirmed the need to fail over, you may override the disabled cmrecovercl command by using the -f forcing option:

```sh
# cmrecovercl -f
```

This should only be used after positive confirmation from the remote site.
Building a Continental Cluster

Switching to the Recovery Packages in Case of Disaster

If the monitored cluster comes back up following an alert or alarm, but you are certain that the primary packages cannot start (say, because of damage to the disks on the primary site), you need to use a special procedure to initiate recovery:

1. Use the \texttt{cmhaltcl} command to halt the primary cluster.
2. Wait for the monitor to send an alert.
3. Use \texttt{cmrecovercl -f} to perform recovery.

After the \texttt{cmrecovercl} command is issued, ContinentalClusters displays a warning message like the following and prompts for a verification that recovery should proceed (the names “LAcluster” and “NYcluster” are examples):

\begin{verbatim}
WARNING: This command will take over for the primary cluster “LAcluster” by starting the recovery package on the recovery cluster “NYcluster”. You must follow your site disaster recovery procedure to ensure that the primary packages on “LAcluster” are not running and that recovery on “NYcluster” is necessary. Continuing with this command while the applications are running on the primary cluster may result in data corruption.

Are you sure that the primary packages are not running and will not come back, and are you certain that you want to start the recovery packages? [Y/N]

Reply Y to proceed only if you are sure that recovery should take place. After replying Y to the prompt, you should see a group of messages like the following as the processing of each recovery group occurs (the message about the data receiver package only appears if you are using logical data replication with data sender and receiver packages):

Processing the recovery group nfsgroup on recovery cluster eastcoast
Disabling switching for data receiver package nfsvsdrepkg on recovery cluster eastcoast
Halting data receiver package nfsvsdrepkg on recovery cluster eastcoast
Starting recovery package nfsvrepkg on recovery cluster eastcoast
Enabling package nfsvrepkg in cluster eastcoast

-----------------
exit status = 0
-----------------
\end{verbatim}

\textbf{NOTE} \ After the \texttt{cmrecovercl} command is issued, there is a delay of at least 90 seconds per recovery group as the command makes sure that the package is not active on another cluster. Please be patient.
Building a Continental Cluster

Switching to the Recovery Packages in Case of Disaster

Use the `cmviewcl` command on the local cluster to confirm that the recovery packages are running correctly. Following recovery, you can halt the package that was monitoring the remote cluster if you wish. If you do not do so, you will continue to receive notification if there is a change in the remote cluster's state. The following table shows the status of Continental Cluster's packages after recovery has taken place, and applications are now running on the local cluster.

### Table 5-6 Status of Continental Clusters Packages After Recovery

<table>
<thead>
<tr>
<th>Data Replication Method</th>
<th>Primary Cluster</th>
<th>Recovery Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Primary Package</td>
<td>Optional Monitor Package</td>
</tr>
<tr>
<td>Physical—Symmetrix</td>
<td>Halted</td>
<td>Not used</td>
</tr>
<tr>
<td>Physical—XP Series</td>
<td>Halted</td>
<td>Not used</td>
</tr>
<tr>
<td>Logical—Oracle</td>
<td>Halted</td>
<td>Not used</td>
</tr>
<tr>
<td>Standby Database</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

How the cmrecovercl Command Works

The `cmrecovery` command uses the configuration file to loop through each defined recovery group. For each group, the command communicates with the monitor package (`ccmonpkg`) and verifies that the remote cluster is unreachable or down, then if there is a data replication package it is halted, and the recovery package is enabled on the Recovery Cluster. The recovery package can then start up on the local cluster on the appropriate node, as determined by the `FAILOVER_POLICY` configured for the package.

The process continues for the next recovery group, even if there are problems with one recovery group. After processing one recovery group, if the command discovers that the local cluster is back up, the command exits, since the alarm or alert state no longer exists. This process keeps both the primary and recovery packages from running on the remote.
Building a Continental Cluster

Switching to the Recovery Packages in Case of Disaster

distributor and local cluster at the same time, which would result in data corruption.

NOTE

If the remote cluster comes back up following a cluster event but the primary packages cannot run, halt the primary cluster with the cmhaltcl command, then issue cmrecovercl with the -f option.
Building a Continental Cluster

Restoring Disaster Tolerance

After a failover to a cluster occurs, restoring disaster tolerance has many challenges, the most significant of which are:

• Restoring the failed cluster
  Depending on the nature of the disaster you may need to create a new cluster, or you may be able to restore the cluster. Steps for each scenario are discussed in the following sections.

• Resynchronizing the data
  To resynchronize the data, you either restore the data to the cluster and continue with the same data replication procedure, or set up data replication to function in the other direction.

The following sections briefly outline some scenarios for restoring disaster tolerance.

**Restore Clusters to their Original Roles**

If the disaster did not destroy the cluster, you can return both clusters to their original roles. To do this:

1. Make sure that both clusters are up and running, with the recovery packages continuing to run on the surviving cluster.

2. On each cluster, stop the ContinentalClusters monitor package if it is still running:
   
   ```
   # cmhaltpkg ccmonpkg
   ```

3. Compare the clusters to make sure their configurations are consistent. Correct any inconsistencies.

4. For each recovery group where the new cluster will run the primary package:
   
   a. Synchronize the data from the disks on the surviving cluster to the disks on the new cluster. This may be time-consuming.
   
   b. Halt the application on the surviving cluster if necessary, and start it on the new cluster.
   
   c. To keep application down time to a minimum, start the primary
5. Restart the monitor using the following command on each cluster:

```
# cmrunpkg ccmonpkg
```

Alternatively, if you have modified the monitoring package configuration, use the following sequence on each cluster to apply the new configuration and start the monitor:

```
# cmapplyconf -P ccmonpkg.config
# cmmodpkg -e ccmonpkg
```


```
# cmviewconcl
```

**Primary Packages Remain on the Surviving Cluster**

Configure the failed cluster as a recovery-only cluster and the surviving cluster as a primary-only cluster. This minimizes the downtime involved with moving the applications back to the restored cluster. It also assumes that the surviving cluster has sufficient resources to handle running all critical applications indefinitely. Use the following procedure:

1. Halt the monitor packages. Issue the following command on each cluster:

```
# cmhaltpkg ccmonpkg
```

2. Edit the ContinentalClusters ASCII configuration file. You will need to change the definitions of monitoring clusters, and switch the names of primary and recovery packages in the definitions of recovery groups. You may also need to re-create data sender and data receiver packages.

3. Check and apply the ContinentalClusters configuration:

```
# cmcheckconcl -v -C cmconcl.config
# cmapplyconcl -v -C cmconcl.config
```

4. Restart the monitor packages. Issue the following command on each cluster:

```
# cmmodpkg -e ccmonpkg
```

5. View the status of the ContinentalCluster.
Newly Created Cluster Will Run Primary Packages

After you create a new cluster to replace the damaged one, you may choose to restore the critical applications to the new cluster and restore the other cluster to its role as backup for the recovered packages.

1. Configure the new cluster as an MC/ServiceGuard cluster. Use the `cmviewconcl` command on the surviving cluster and compare the results to the new cluster configuration. Correct any inconsistencies on the new cluster.

2. Halt the monitor package on the surviving cluster:

   ```
   # cmhaltpkg ccmonpkg
   ```

3. Edit the continental cluster configuration file to replace the data from the old failed cluster with data from the new cluster. Check and apply the ContinentalClusters configuration:

   ```
   # cmcheckconcl -v -C cmconcl.config
   # cmapplyconcl -v -C cmconcl.config
   ```

4. For each recovery group where the new cluster will run the primary package:
   a. Synchronize the data from the disks on the surviving cluster to the disks on the new cluster. This may be time-consuming.
   b. Halt the application on the surviving cluster if necessary, and start it on the new cluster.
   c. To keep application down time to a minimum, start the primary package on the cluster before resynchronizing the data of the next recovery group.

5. If the new cluster acts as recovery cluster for any recovery group, create a monitor package for the new cluster.

   Use the following command to apply the configuration of the new monitor package:

   ```
   # cmapplyconf -p ccmonpkg.config
   ```

6. Restart the monitor package on the surviving cluster:
# cmrunpkg ccmONpkg


# cmviewconcl

Newly Created Cluster Will Function as Recovery Cluster for All Recovery Groups

After you replace the failed cluster, if you are concerned with the
downtime involved in moving the applications back, you can change the
surviving cluster to the role of primary cluster for all recovery groups,
and configure the new cluster as a recovery cluster for all those groups.

You would configure the new cluster as a standard MC/ServiceGuard
dcluster, and follow the usual procedure to configure the continental
custer with the new cluster used as a recovery cluster for all recovery
groups.
Maintaining a Continental Cluster

The following common maintenance tasks are described in this section:

- Adding a Node to a Cluster or Removing a Node from a Cluster
- Adding a Package to a Continental Cluster
- Removing a Package from the Continental Cluster
- Changing Monitoring Definitions
- Checking the Status of Clusters, Nodes and Packages
- Reviewing Log Files
- Renaming a Continental Cluster
- Deleting a Continental Cluster configuration
- Checking Java Versions

CAUTION

You should never issue the `cmrunpkg` command for a recovery package when ContinentalClusters is enabled, because there is no sure way of preventing a package that is running on the one cluster from running on the other cluster if the package is started using this command. The potential for data corruption is great.

Adding a Node to a Cluster or Removing a Node from a Cluster

To add a node to or remove a node from the continental cluster, use the following procedure:

1. Halt any monitor packages that are running by issuing the following command on both clusters:
   
   ```
   # cmhalt pkg ccmon pkg
   ```

2. Add or remove the node in a cluster by editing the ServiceGuard cluster configuration file and applying the configuration:
   
   ```
   # cmapplyconf -C cluster.config
   ```
3. Edit the ContinentalClusters configuration ASCII file to add or remove the node in the cluster.

4. For added nodes, ensure the the /etc/cmcluster/cmomhosts file is set up correctly on the new node. Refer to “Preparing Security Files” on page 159. Ensure that the /.rhosts file on all nodes (including the new node) contains an entry allowing write access by the host on which you are running the configuration commands.

5. Check and apply the configuration using the cmcheckconcl and cmapplyconcl commands.

6. Remove entries from the /.rhosts file if desired.

7. Restart the monitor packages on both clusters.

8. View the status of the continental cluster.

```cmviewconcl```

**Adding a Package to the Continental Cluster**

To add a new package for possible recovery to the ContinentalClusters configuration, you must first configure a new primary package and recovery package, then you must add a new recovery group to the ContinentalClusters configuration file. In addition, you must ensure that data replication is provided for the new package either through hardware or software.

Adding a new package does not require you to bring down either cluster. However, in order to implement the new configuration, you must

1. Configure the new primary and recovery packages by editing the new package configuration files and control scripts.

2. Use the ServiceGuard cmapplyconf command to add the primary package to one cluster, and the recovery package to the other cluster.

3. Provide the appropriate data replication for the new package.

4. Create the new recovery group in the ContinentalClusters configuration file.

5. Ensure that the /.rhosts file on all nodes contains an entry allowing write access by the host on which you are running the configuration commands.

6. Halt the monitor packages on both clusters.
Building a Continental Cluster

Maintaining a Continental Cluster

7. Use the `cmapplyconcl` command to apply the new ContinentalClusters configuration.
8. Restart the monitor packages on both clusters.
9. View the status of the continental cluster.

```cmviewconcl```

Removing a Package from the Continental Cluster

To remove a package from the ContinentalClusters configuration, you must first remove the recovery group from the ContinentalClusters configuration file. Removing the package does not require you to bring down either cluster. However, in order to implement the new configuration, you must

1. Edit the continental clusters configuration file, deleting the recovery group.
2. Halt the monitor packages on both clusters.
3. Use the `cmapplyconcl` command to apply the new ContinentalClusters configuration.
4. Restart the monitor packages on both clusters.
5. Use the ServiceGuard `cmdeleteconf` command to remove each package in the recovery group.
6. View the status of the continental cluster.

```cmviewconcl```

Changing Monitoring Definitions

You can change the monitoring definitions in the configuration without bringing down either cluster. This includes adding, removing, or changing the cluster events; changing the timings; and adding, removing, or changing the notification messages.

Use the following steps:

1. Edit the continental clusters configuration file to incorporate the new or changed monitoring definitions.
2. Halt the monitor packages on both clusters.
Building a Continental Cluster
Maintaining a Continental Cluster

3. Ensure that the `.rhosts` file on all nodes (including the new node) contains an entry allowing write access by the host on which you are running the configuration commands.

4. Use the `cmapplyconcl` command to apply the new configuration.

5. Restart the monitor packages on both clusters.

6. View the status of the continental cluster.
   ```bash
   # cmviewconcl
   ``

### Checking the Status of Clusters, Nodes, and Packages

To check on the status of the continental clusters and associated packages, use the `cmviewconcl` command. The command lists the status of the clusters, associated package status, and configured events status.

The following is an example of `cmviewconcl` output in a situation where there is a single recovery group for which the primary cluster is cjc838 and the recovery cluster is cjc1234.

```bash
#cmviewconcl

WARNING: Primary cluster cjc838 is in an alarm state
(cmrecovercl is enabled on recovery cluster cjc1234)

CONTINENTAL CLUSTER  cjccc1
RECOVERY CLUSTER  cjc1234

PRIMARY CLUSTER STATUS EVENT LEVEL POLLING INTERVAL
   cjc838  down   ALARM        20

PACKAGE RECOVERY GROUP  prg1
   PACKAGE           ROLE    STATUS
       cjc838/primary  primary  down
       cjc1234/recovery recovery  up
```

The following is an example of `cmviewconcl` output from a primary cluster that it down.

```bash
persian (root 2131): cmviewconcl -v
WARNING: Primary cluster cjc838 is in an alarm state
(cmrecovercl is enabled on recovery cluster cjc1234)

Primary cluster cjc838 is not configured to monitor recovery cluster cjc1234
```
## Building a Continental Cluster

### Maintaining a Continental Cluster

<table>
<thead>
<tr>
<th>CONTINENTAL CLUSTER</th>
<th>cjjccc1</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECOVERY CLUSTER</td>
<td>cjc1234</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PRIMARY CLUSTER</th>
<th>STATUS</th>
<th>EVENT LEVEL</th>
<th>POLLING INTERVAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>cjc838</td>
<td>down</td>
<td>ALARM</td>
<td>20</td>
</tr>
</tbody>
</table>

### CONFIGURED EVENT STATUS

<table>
<thead>
<tr>
<th>CONFIGURED EVENT</th>
<th>STATUS</th>
<th>DURATION</th>
<th>LAST NOTIFICATION SENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>alert</td>
<td>unreachable</td>
<td>15 sec</td>
<td>--</td>
</tr>
<tr>
<td>alarm</td>
<td>unreachable</td>
<td>30 sec</td>
<td>--</td>
</tr>
<tr>
<td>alarm</td>
<td>down</td>
<td>0 sec</td>
<td>Fri May 12 12:13:06 PDT 2000</td>
</tr>
<tr>
<td>alert</td>
<td>error</td>
<td>0 sec</td>
<td>--</td>
</tr>
<tr>
<td>alert</td>
<td>up</td>
<td>20 sec</td>
<td>--</td>
</tr>
<tr>
<td>alert</td>
<td>up</td>
<td>40 sec</td>
<td>--</td>
</tr>
</tbody>
</table>

### PACKAGE RECOVERY GROUP prg1

<table>
<thead>
<tr>
<th>PACKAGE</th>
<th>ROLE</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>cjc838/primary</td>
<td>primary</td>
<td>down</td>
</tr>
<tr>
<td>cjc1234/recovery</td>
<td>recovery</td>
<td>up</td>
</tr>
</tbody>
</table>

The following is the output of a `cmviewconcl` command displaying data for a mutual recovery configuration in which each cluster has both the primary and the recovery roles—the primary role for one recovery group and the recovery role for the other recovery group:

### CONTINENTAL CLUSTER ccluster1

<table>
<thead>
<tr>
<th>RECOVERY CLUSTER</th>
<th>PTST_dts1</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>PRIMARY CLUSTER</th>
<th>STATUS</th>
<th>EVENT LEVEL</th>
<th>POLLING INTERVAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTST_sanfran</td>
<td>Unmonitored</td>
<td>unmonitored</td>
<td>1 min</td>
</tr>
</tbody>
</table>

### CONFIGURED EVENT STATUS

<table>
<thead>
<tr>
<th>CONFIGURED EVENT</th>
<th>STATUS</th>
<th>DURATION</th>
<th>LAST NOTIFICATION SENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>alert</td>
<td>unreachable</td>
<td>1 min</td>
<td>--</td>
</tr>
<tr>
<td>alert</td>
<td>unreachable</td>
<td>2 min</td>
<td>--</td>
</tr>
<tr>
<td>alarm</td>
<td>unreachable</td>
<td>3 min</td>
<td>--</td>
</tr>
<tr>
<td>alert</td>
<td>down</td>
<td>1 min</td>
<td>--</td>
</tr>
<tr>
<td>alert</td>
<td>down</td>
<td>2 min</td>
<td>--</td>
</tr>
<tr>
<td>alarm</td>
<td>down</td>
<td>3 min</td>
<td>--</td>
</tr>
<tr>
<td>alert</td>
<td>error</td>
<td>0 sec</td>
<td>--</td>
</tr>
<tr>
<td>alert</td>
<td>up</td>
<td>1 min</td>
<td>--</td>
</tr>
</tbody>
</table>

### RECOVERY CLUSTER PTST_dts1

<table>
<thead>
<tr>
<th>PRIMARY CLUSTER</th>
<th>STATUS</th>
<th>EVENT LEVEL</th>
<th>POLLING INTERVAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTST_dts1</td>
<td>Unmonitored</td>
<td>unmonitored</td>
<td>1 min</td>
</tr>
</tbody>
</table>

### CONFIGURED EVENT STATUS

<table>
<thead>
<tr>
<th>CONFIGURED EVENT</th>
<th>STATUS</th>
<th>DURATION</th>
<th>LAST NOTIFICATION SENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>alert</td>
<td>unreachable</td>
<td>1 min</td>
<td>--</td>
</tr>
<tr>
<td>alert</td>
<td>unreachable</td>
<td>2 min</td>
<td>--</td>
</tr>
<tr>
<td>alarm</td>
<td>unreachable</td>
<td>3 min</td>
<td>--</td>
</tr>
</tbody>
</table>
Building a Continental Cluster

Maintaining a Continental Cluster

For a more comprehensive status of component clusters, nodes, and packages, use the `cmviewcl` command on both clusters. On each cluster, you should make note of which nodes the primary packages are running on, as well as data sender and data receiver packages, if you are using them for logical data replication. You should verify that the monitor is running on each cluster on which it is configured.

The following is an example of `cmviewcl` output for a cluster (`nycluster`) running a monitor package. Note that the recovery package `salespkg_bak` is not running, and is shown as an unowned package. This is the expected display while the other cluster is running `salespkg`.

```
<table>
<thead>
<tr>
<th>CLUSTER</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>nycluster</td>
<td>up</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NODE</th>
<th>STATUS</th>
<th>STATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>nynode1</td>
<td>up</td>
<td>running</td>
</tr>
</tbody>
</table>

Network_Parameters:

<table>
<thead>
<tr>
<th>INTERFACE</th>
<th>STATUS</th>
<th>PATH</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRIMARY</td>
<td>up</td>
<td>12.1</td>
<td>lan0</td>
</tr>
<tr>
<td>PRIMARY</td>
<td>up</td>
<td>56.1</td>
<td>lan1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NODE</th>
<th>STATUS</th>
<th>STATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>nynode2</td>
<td>up</td>
<td>running</td>
</tr>
</tbody>
</table>

Network_Parameters:

<table>
<thead>
<tr>
<th>INTERFACE</th>
<th>STATUS</th>
<th>PATH</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRIMARY</td>
<td>up</td>
<td>4.1</td>
<td>lan0</td>
</tr>
<tr>
<td>PRIMARY</td>
<td>up</td>
<td>56.1</td>
<td>lan1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PACKAGE</th>
<th>STATUS</th>
<th>STATE</th>
<th>PKG_SWITCH</th>
<th>NODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ccmonpkg</td>
<td>up</td>
<td>running</td>
<td>enabled</td>
<td>nynode2</td>
</tr>
</tbody>
</table>

Script_Parameters:

<table>
<thead>
<tr>
<th>ITEM</th>
<th>NAME</th>
<th>STATUS</th>
<th>MAX_RESTARTS</th>
<th>RESTARTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service</td>
<td>ccmonpkg.srv</td>
<td>up</td>
<td>20</td>
<td>0</td>
</tr>
</tbody>
</table>
```
Building a Continental Cluster

Maintaining a Continental Cluster

Node_Switching_Parameters:

<table>
<thead>
<tr>
<th>NODE_TYPE</th>
<th>STATUS</th>
<th>SWITCHING</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary</td>
<td>up</td>
<td>enabled</td>
<td>nynode2  (current)</td>
</tr>
<tr>
<td>Alternate</td>
<td>up</td>
<td>enabled</td>
<td>nynode1</td>
</tr>
</tbody>
</table>

UNOWNED PACKAGES

<table>
<thead>
<tr>
<th>PACKAGE</th>
<th>STATUS</th>
<th>STATE</th>
<th>PKG_SWITCH</th>
<th>NODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>salespkg_bak</td>
<td>down</td>
<td></td>
<td></td>
<td>unowned</td>
</tr>
</tbody>
</table>

Policy_Parameters:

<table>
<thead>
<tr>
<th>POLICY_NAME</th>
<th>CONFIGURED_VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failover</td>
<td>unknown</td>
</tr>
<tr>
<td>Failback</td>
<td>unknown</td>
</tr>
</tbody>
</table>

Script_Parameters:

<table>
<thead>
<tr>
<th>ITEM</th>
<th>STATUS</th>
<th>NODE_NAME</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subnet</td>
<td>unknown</td>
<td>nynode1</td>
<td>195.14.171.0</td>
</tr>
<tr>
<td>Subnet</td>
<td>unknown</td>
<td>nynode2</td>
<td>195.14.171.0</td>
</tr>
</tbody>
</table>

Node_Switching_Parameters:

<table>
<thead>
<tr>
<th>NODE_TYPE</th>
<th>STATUS</th>
<th>SWITCHING</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary</td>
<td>down</td>
<td></td>
<td>nynode1</td>
</tr>
<tr>
<td>Alternate</td>
<td>down</td>
<td></td>
<td>nynode2</td>
</tr>
</tbody>
</table>

You can also use the `ps` command to check for the status of the ContinentalClusters monitor daemons cmclrmond and cmclsentryd. They should be running on the cluster node where the monitor package is running.

Reviewing Messages and Log Files

The ContinentalClusters commands—cmquerycl, cmcheckconcl, cmapplyconcl, and cmrecovercl—all display messages on the standard output, which is the first place to look for error messages.

```
>------------ Event Monitoring Service Event Notification ------------<

Notification Time: Wed Nov 10 21:00:39 1999

system1 sent Event Monitor notification information:
/cluster/concl/ccluster1/clusters/LAclust/status/unreachable is = 15.

User Comments:
Cluster "LAclust" has status "unreachable" for 15 sec

>---------- End Event Monitoring Service Event Notification ----------<
```

All notification messages associated with cluster events are reported in /etc/cmcluster/cmconcl/eventlog on the cluster where monitoring is taking place. An example of output from this file follows:
Building a Continental Cluster

Maintaining a Continental Cluster

In addition, if you have defined a TEXTLOG destination, notification messages are sent to the file you specified. (See “Editing Section 3—Monitoring Definitions” on page 170 for more information.)

You can also review the monitor startup and shutdown log file
/etc/cmcluster/ccmonpkg/ccmonpkg.cntl.log on any node where a ContinentalClusters monitor has been running. Information about the primary or recovery packages may be found in their respective startup and shutdown log files.

Messages from the ContinentalClusters daemon are reported in log file
/var/adm/cmconcl/sentryd.log, and Object Manager messages appear in /var/opt/cmom/cmomd.log. These messages may be helpful in troubleshooting. Use the cmreadlog command to view the entries in these files. Examples:

```
#!/opt/cmom/tools/bin/cmreadlog -f
/var/adm/cmconcl/sentryd.log slog.txt

#!/opt/cmom/tools/bin/cmreadlog -f /var/opt/cmom/cmomd.log omlog.txt
```

The following is sample output from the cmreadlog command for the sentryd.log file:

```
Oct 20 18:28:22:[main,5,main]]:FATAL:dr.sentryd:No continental cluster found on this node
Oct 22 13:38:45:[[Thread-309,5,main]]:ERROR:dr.sentryd:Error connecting to axe28
Oct 22 13:38:45:[[Thread-309,5,main]]:ERROR:dr.sentryd:Connection refused
Oct 22 13:38:45:[[Thread-309,5,main]]:INFO:dr.sentryd:Connection failed to axe28
Oct 22 13:38:45:[[Thread-311,5,main]]:ERROR:dr.sentryd:Cannot find cluster KC-cluster at location axe29
Oct 22 13:38:45:[[Thread-311,5,main]]:ERROR:dr.sentryd:null result from query
```

General information about MC/ServiceGuard operation is found in
/var/adm/syslog/syslog.log.

Deleting a Continental Cluster Configuration

The cmdeleteconcl command is used to delete the configuration on all nodes in the continental cluster configuration. To delete a continental cluster use the following command:

```
#cmdeleteconcl
```

This command deletes the ContinentalCluster configuration.
NOTE

If you are modifying the configuration, you simply re-issue the cmapplyconcl command. There is no need to delete the previous configuration.

Renaming a Continental Cluster

To rename an existing continental cluster, first remove the old continental cluster configuration using the cmdeleteconcl command, then edit the ASCII configuration file to include the new name, and finally reapply the configuration using the cmapplyconcl command. To remove continental clusters configuration, use the following command:

```
# cmdeleteconcl
```

You can now edit the CONTINENTAL_CLUSTER_NAME field in the configuration ASCII file, and run the cmapplyconcl command to configure the continental cluster with a new name.

Checking Java Versions

Some components of ContinentalClusters are executed from Java .jar files. To obtain version information about these files, use the what.sh script provided in the /opt/cmconcl/jar directory. Example:

```
# /opt/cmconcl/jar/what.sh configcl.jar
```
Physical Data Replication for ContinentalClusters Using Continuous Access XP

6 Physical Data Replication for ContinentalClusters Using Continuous Access XP

This chapter shows how to use the MetroCluster/CA package control script template to set up data replication between the primary and secondary packages in a continental cluster that uses the HP SureStore E Disk Array XP Series with Continuous Access XP. These packages run on different clusters within the continental cluster.

This chapter has the following sections:

- Overview
- Preparing the Continental Cluster for Data Replication
- Creating and Exporting Volume Groups
- Setting up a Primary Package on the Primary Cluster
- Setting up a Recovery Package on the Recovery Cluster
- Setting up the Continental Cluster Configuration
- Switching to the Recovery Cluster in Case of Disaster
- Failback Scenarios
- Maintaining the Continuous Access XP Data Replication Environment

It is assumed that readers are already familiar with MC/ServiceGuard configuration tasks, continental cluster concepts and installation procedures, HP SureStore E Disk Array XP Series and Continuous Access XP concepts and Raid Manager configuration and use.

NOTE

This chapter is for ContinentalClusters users only. Users of MetroCluster with Continuous Access XP should not use the procedures in this chapter. Refer instead to Chapter 3, “Building a Metropolitan Cluster Using MetroCluster/CA.”
Overview

Physical data replication with the HP SureStore E Disk Array XP Series uses the Continuous Access XP package control script template, which is installed in the /opt/cmcluster/toolkit/SGCA directory when you install ContinentalClusters. This directory includes the following files:

- sgcapkg.cntl—the Continuous Access XP package control script template. This template must be customized for the specific XP Disk Array and HP 9000 host system configuration. Copies of this template must be customized for each separate MC/ServiceGuard package and for each MC/ServiceGuard Cluster.

- Samples-CC—a directory containing sample convenience shell scripts that must be edited before using. These shell scripts may help to automate some configuration tasks. These scripts are contributed, and not supported. This directory also contains example Raid Manager configuration files and other examples.

The Continuous Access XP package control script template should be used on continental cluster nodes where the following are true:

- Data are on an HP SureStore E XP series disk array.
- Data are replicated to a second XP series disk array using the Continuous Access XP facility.

To use Continuous Access XP data replication in a continental cluster, the following must also be installed and configured on each HP 9000 host system that might execute the application:

- HP-UX 11.0 or greater
- HP ContinentalClusters A.01.00 or greater
- Raid Manager XP host-based software for control and status of the XP series disk arrays.
- MC/ServiceGuard A.11.09 or greater

The procedures in this chapter assume that you are configuring one or more ServiceGuard application packages whose data reside on the XP disks and are replicated between the two XP disk arrays with Continuous Access XP. The procedures must be repeated for each application package.
Continuous Access can be configured in either synchronous or asynchronous mode. Asynchronous mode is enabled by setting the `FENCE` parameter to `ASYNC` in the package configuration ASCII file.

While the cluster is running, all the XP devices (LUNs) that belong to the same package and are defined in a single Raid Manager device group must be in the same state at the same time. Manual changes of these states can cause the package to halt or to not start up properly due to unexpected conditions. In general, it is recommended that no manual changes of state be performed while the package and cluster are running, except during the repair process. (For example, it is acceptable to run `pairresync` to recover from a complete CA link failure. However, if manual changes are made during the repair process, devices must be returned to their original states afterwards.)

---

**WARNING**  
The logic of the special control script always assumes that a given PVOL/SVOL state combination was not due to manual configuration changes.
Preparing the Continental Cluster for Data Replication

These procedures will prepare your MC/ServiceGuard clusters for use with Continuous Access XP data replication in a continental cluster.

1. Ensure that the XP Series disk arrays are correctly cabled to each host system in the two clusters that will run packages whose data reside on the arrays.

   Each XP Series disk array must be configured with redundant ESCON links, each of which is connected to a different LCP or RCP card. To prevent a single point of failure (SPOF), there must be at least two physical boards in each XP for the ESCON links. Each board usually has multiple ports. However, a redundant ESCON link must be connected to a port on a different physical board from the board that has the primary ESCON link. When using bi-directional configurations, where data center A backs up data center B and data center B backs up data center A, you must have at least four ESCON links, two in each direction. Four ESCON links are also required in uni-directional configurations in which you want to allow failback.

2. Install the Raid Manager XP software on each host system that has data residing on the XP disk arrays.

3. Edit the /etc/services file, adding an entry for the Raid Manager instance to be used with ContinentalClusters.

   The format of the entry is:
   
   horcm<instance-number>  <port-number>/udp

   For example:
   
   horcm0  11000/udp    #Raid Manager instance 0

   For more detail, see the
   
   /opt/cmcluster/toolkit/SGCA/Samples-CC/services.example
   
   file.

4. Use the ioscans command to determine what devices on the XP disk array have been configured as command devices. The device-specific information in the rightmost column of the ioscans output will have the suffix -CM for these devices; e.g., OPEN-3-CM.
Preparing the Continental Cluster for Data Replication

If there are no configured command devices on the disk array, you must create two before proceeding. Each command device must have alternate links (PV Links). The first command device is the primary command device. The second command device is a redundant command device and is used only upon failure of the primary command device. The command devices must be mapped to the various host interfaces by using the SVP (disk array console) or a remote console.

5. Copy the default Raid Manager configuration file to an instance-specific name. Example:
   ```
   # cp /etc/horcm.conf /etc/horcm0.conf
   ```

6. Create a minimum Raid Manager configuration file by editing the following fields in the file created in the previous step:
   - `HORCM_MON`—enter the host-name of the system on which you are editing and the TCP/IP port number specified for this Raid Manager instance in the `/etc/services` file.
   - `HORCM_CMD`—enter the primary and alternate link device file names for both the primary and redundant command devices (for a total of four raw device file names).

   **WARNING**
   Make sure that the redundant command device is NOT on the same physical device as the primary command device. Also, make sure that it is on a different bus inside the XP series disk array.

7. Start the Raid Manager instance by using the command `horcmstart.sh <instance-#>`. For example:
   ```
   # horcmstart.sh 0
   ```

8. Export the environment variable that specifies the Raid Manager instance to be used by the Raid Manager commands. For example, with the POSIX shell type:
   ```
   # export HORCMINST=0
   ```

   Now, you can use Raid Manager commands to get further information from the disk arrays.

   To verify the software revision of the Raid Manager and the firmware
Preparing the Continental Cluster for Data Replication

revision of the XP disk array, use the command:

```
# raidqry -l
```

The Raid Manager software must be at least revision 01.02.03 and the firmware must be at least revision C.

To get a list of the available devices on the disk arrays, use the command

```
# raidscan
```

This command must be invoked separately for each host interface connection to the disk array. For example, if there are two Fibre Channel host adapters, you might use the commands:

```
# raidscan -p CL1-A
# raidscan -p CL1-B
```

---

**NOTE**

There must also be alternate links for each device, and these alternate links must be on different busses inside the XP disk array. These alternate links, for example, may be CL2-E and CL2-F.

---

Unless the devices have been previously paired either on this or another host, the devices will show up as SMPL (simplex). Paired devices will show up as PVOL (primary volume) or SVOL (secondary volume).

9. Determine which devices will be used by the application package. Define a device group that contains all of these devices. It is recommended that you use a name that is easily associated with the package. For example, a device group name of “db-payroll” is easily associated with the database for the payroll application. A device group name of “group1” would be more difficult to easily relate to an application.

Edit the Raid Manager configuration file (horcm0.conf in the above example) to include the devices and device group used by the application package. Only one device group may be specified for all of the devices that belong to a single application package. These devices are specified in the field `HORCM_DEV`.

Also complete the `HORCM_INST` field, supplying the names of only those hosts that are attached to the XP disk array that is remote from
Preparing the Continental Cluster for Data Replication

the disk array directly attached to this host. For example, with the continental cluster shown in Figure 6-1 (node 1 and node 2 in the primary cluster and nodes 3 and 4 in the recovery cluster), you would specify only nodes 3 and 4 in the HORCM_INST field in a file you are creating on node 1 on the primary cluster. Node 1 would have previously been specified in the HORCM_MON field.

Figure 6-1 Continental Cluster

10. Restart the Raid Manager instance so that the new information in the configuration file is read. Use the commands:

```
# horcmshutdown.sh <instance-#>
# horcmstart.sh <instance-#>
```

11. Repeat these steps on each host that will run this particular application package. If a host may run more than one application package, you must incorporate device group and host information for each of these packages. Note that the Raid Manager configuration file must be different for each host, especially for the HORC_MON and HORC_INST fields.

12. If not previously done, you can create the paired volumes by using the command:

```
# paircreate -g <devgroup> -f <fencelvel> -v1 -c15
```
Preparing the Continental Cluster for Data Replication

This command must be issued before you create volume groups.

**WARNING**

**Paired devices must be of compatible sizes and types.**

When using the `paircreate` command to create PVOL/SVOL CA pairs, specify the `-c 15` switch to ensure the fastest data copy from PVOL to SVOL.

**Notes on the Raid Manager Configuration**

A single XP device group must be defined for each package on each host that is connected to the XP series disk array. Device groups are defined in the Raid Manager configuration file under the heading "HORCM_DEV".

The disk target IDs and LUNs for all Physical Volumes (PVs) defined in Volume Groups (VGs) that belong to the package must be defined in one XP device group on each host system that may ever run one or more ContinentalClusters packages. The device group name (dev_group) is user-defined and must be the same on each host in the continental cluster that accesses the XP disk array. The device group name (dev_group) must be unique within the cluster; it should be a name that is easily associated with the application name or MC/ServiceGuard package name.

The device name (dev_name) is user-defined and must be the same on each host in the continental cluster that accesses the XP disk array. The device name (dev_name) must be unique among all devices in the cluster.

The TargetID and LU# fields for each device name may be different on different hosts in the clusters, to allow for different hardware I/O paths on different hosts. See the sample convenience scripts in the Samples-CC directory included with this toolkit for examples.
Creating and Exporting Volume Groups

Use the following procedure to create and export volume groups.

1. Define the appropriate Volume Groups on each host system that might run the application package. Use the commands:

   ```bash
   # mkdir /dev/vgxx
   # mknod /dev/vgxx/group c 64 0xnn0000
   ``

   where the name /dev/vgxx and the number nn are unique within the entire cluster.

2. Create the Volume Group only on the primary system on the primary cluster. Use the `vgcreate` and perhaps the `vgextend` command, specifying the appropriate special device file names. See the sample script `Samples-CC/mk1VGs`.

3. Create the logical volume(s) for the volume group.

4. Export the VGs on the primary system without removing the special device files. Use the commands:

   ```bash
   # vgchange -a n <vgname>
   # vgexport -s -p -m <mapfilename> <vgname>
   ``

   Make sure that you copy the mapfiles to all of the host systems:

   ```bash
   # vgchange -a y <vgname>
   # vgcfgbackup <vgname>
   # vgchange -a n <vgname>
   ``

   See the sample script `Samples-CC/mk2imports`.

5. On the primary cluster import the VGs on all of the other systems that might run the MC/ServiceGuard package and backup the LVM configuration. Use the commands:

   ```bash
   # vgimport -s -m <mapfilename> <vgname>
   # vgchange -a y <vgname>
   # vgcfgbackup <vgname>
   # vgchange -a n <vgname>
   ``

   See the sample script `Samples-CC/mk2imports`.

6. On the recovery cluster import the VGs on all of the systems that might run the MC/ServiceGuard recovery package and backup the LVM configuration. Use the commands:

   ```bash
   # pairsplit -g <dev_name> -rw
   ```
Physical Data Replication for ContinentalClusters Using Continuous Access XP

Creating and Exporting Volume Groups

```
# vgimport -s -m <mapfilename> <vgname>
# vgchange -a y <vgname>
# vgcfgbackup <vgname>
# vgchange -a n <vgname>
# pairresync -g <dev_name> -c 15
```

See the sample script `Samples-CC/mk2imports`.

You can skip the `pairsplit/pairresync`, but you will not be able to activate the volume group to perform the `vgcfgbackup`. Perform the `vgcfgbackup` when the volume group is activated during the first recovery package activation.

When using the `pairresync` command to resynchronize PVOL/SVOL CA pairs, specify the `-c 15` switch to ensure the fastest resynchronization which reduces the vulnerability to a rolling disaster.
Setting up a Primary Package on the Primary Cluster

Use the procedures in this section to configure a primary package on the primary cluster. Consult the MC/ServiceGuard documentation for more detailed instructions on setting up MC/ServiceGuard with packages, and for instructions on how to start, halt, and move packages and their services between nodes in a cluster.

NOTE

Neither the primary cluster nor the recovery cluster may configure an XP series paired volume, PVOL or SVOL, as a cluster lock disk. A cluster lock disk must always be writeable. Since it cannot be guaranteed that either half of a paired volume is always writeable, neither half may be used as a cluster lock disk. A configuration with a cluster lock disk that is part of a paired volume is not a supported configuration.

1. Create and test a standard MC/ServiceGuard cluster using the procedures described in the user’s manual, Managing MC/ServiceGuard.

2. Install ContinentalClusters on all the cluster nodes in the primary cluster (Skip this step if the software has been preinstalled)

NOTE

MC/ServiceGuard should already be installed on all the cluster nodes.

Run swinstall(1m) to install ContinentalClusters product from an SD depot. The toolkit control file and contributed scripts will reside in the /opt/cmcluster/toolkit/SGCA directory. One fileset contains the Continuous Access (CA) template control file and sample scripts directory.

3. When swinstall(1m) has completed, create a directory as follows for the new package in the primary cluster:

```
# mkdir /etc/cmcluster/<package_name>
```

Create an MC/ServiceGuard package configuration file in the primary cluster with the commands:
Physical Data Replication for ContinentalClusters Using Continuous Access XP

Setting up a Primary Package on the Primary Cluster

```bash
# cd /etc/cmcluster/<package_name>
# cmmakepkg -p <package_name>.ascii
```

Customize it as appropriate to your application. Be sure to include the pathname of the control script (/etc/cmcluster/<package_name>/<package_name>.cntl) for the `RUN_SCRIPT` and `HALT_SCRIPT` parameters.

Set the `AUTO_RUN` flag (`PKG_SWITCHING_ENABLED` in ServiceGuard A.11.09) to `NO`. This is to ensure the package will not start when the cluster starts.

Only after primary packages start, use `cmmodpkg` to enable package switching on all primary packages. Enabling package switching in the package configuration would automatically start the primary package when the cluster starts. However, had there been a primary cluster disaster, resulting in the recovery package starting and running on the recovery cluster, the primary package should not be started until after first stopping the recovery package.

4. Copy the Continuous Access XP package control script to this new directory and rename it to `<package_name>.cntl`. Note that each package has its own directory and its own uniquely named control script.

```
NOTE
Do not create an MC/ServiceGuard package control script template with the `cmmakepkg -s` command or by using SAM. The package control script template included with this product should be used instead.
```

The scripts in the `Samples-CC` directory are examples of ways to automate some of the tasks. These scripts are for your convenience and are not supported. These example scripts should be copied to another directory (e.g., `/etc/cmcluster/SGCA`) and then edited to customize them for the specific environment.

5. Edit the package control script `<package_name>.cntl` as follows:

   a. If necessary, add the path where the Raid Manager software binaries have been installed to the `PATH` environment variable. The usual location is currently `/usr/bin`. In this case, you do not have to modify the `PATH` variable.

   b. Uncomment behavioral configuration environment variables
Physical Data Replication for Continental Clusters Using Continuous Access XP

Setting up a Primary Package on the Primary Cluster

AUTO_. It is recommended that you retain the default values of these variables unless you have a specific business requirement to change them. See the template for an explanation of these variables.

c. Uncomment the PKGDIR variable and set it to the full path name of the directory where the control script has been placed. This directory must be unique for each package and is used for status data files. For example, set PKGDIR to /etc/cmcluster/<package_name>.

d. Uncomment the DEVICE_GROUP variable and set it to the Raid Manager device group name specified in the Raid Manager configuration file.

e. Uncomment the HORCMINST variable and set it to the Raid Manager instance name used by the continental cluster.

f. Uncomment the FENCE variable and set it to ASYNC, NEVER or DATA according to your business requirements. This variable is used for two purposes:

1. to compare with the actual fence level returned by the array
2. to set the fence level when paircreate operations are performed

See “Setting Fence Levels” on page 65 for more information, including a discussion of asynchronous operation.

g. Uncomment the DC1HOST array variable and set the elements to the hostnames of the primary cluster. The order of the hostnames is not important.

h. Leave commented the DC2HOST array variable.

i. Uncomment the CLUSTER_TYPE variable and set it to continental.

6. Edit the remaining control script variables (VG, LV, FS, IP, SUBNET, SERVICE_NAME, SERVICE_CMD and SERVICE_RESTART) according to the needs of the application as it runs on the primary cluster. See the MC/ServiceGuard manual for more information on these variables.

7. Add customer-defined run and halt commands in the appropriate places according to the needs of the application. See the MC/ServiceGuard manual for more information on these functions.

8. Distribute Continuous Access XP package control script files to other nodes in the primary cluster by using ftp or rcp:
Physical Data Replication for ContinentalClusters Using Continuous Access XP

Setting up a Primary Package on the Primary Cluster

```
# rcp -p /etc/cmcluster/<pkg_name>/<pkg_name>.cntl \
other_node:/etc/cmcluster/<pkg_name>/<pkg_name>.cntl
```

When using `ftp`, be sure to make the file executable on any destination systems.

9. Apply the MC/ServiceGuard configuration using the `cmapplyconf` command or SAM.

10. Verify that each host in the continental cluster has the following files in the directory `/etc/cmcluster/<pkg_name>`:

- `<pkg_name>.cntl`—Continuous Access XP package control script
- `<pkg_name>.ascii` (MC/ServiceGuard package ASCII config file)
- `<pkg_name>.sh` (Package monitor shell script, if applicable)

11. Edit the file `/etc/rc.config.d/raidmgr`, specifying the Raid Manager instance to be used for ContinentalClusters, and specify that the instance be started at boot time.

   The appropriate Raid Manager instance used by ContinentalClusters must be running before the package is started. This normally means that the Raid Manager instance must be started before MC/ServiceGuard is started.

12. Using standard MC/ServiceGuard commands (`cmruncl`, `cmhaltcl`, `cmrunpkg`, `cmhaltpkg`), test the primary cluster for cluster and package startup and package failover.

13. Any running package on the primary cluster that will have a counterpart on the recovery cluster must be halted at this time.
Setting up a Recovery Package on the Recovery Cluster

Use the procedures in this section to configure a recovery package on the recovery cluster. Consult the MC/ServiceGuard documentation for more detailed instructions on setting up MC/ServiceGuard with packages, and for instructions on how to start, halt, and move packages and their services between nodes in a cluster.

**NOTE**

Neither the primary cluster nor the recovery cluster may configure an XP series paired volume, PVOL or SVOL, as a cluster lock disk. A cluster lock disk must always be writable. Since it cannot be guaranteed that either half of a paired volume is always writable, they may not be used as a cluster lock disk. Using a disk as a cluster lock disk that is part of a paired volume is not a supported configuration.

1. Create and test a standard MC/ServiceGuard cluster using the procedures described in the user's manual, Managing MC/ServiceGuard.

2. Install ContinentalClusters on all the cluster nodes in the primary cluster (Skip this step if the software has been preinstalled)

**NOTE**

MC/ServiceGuard should already be installed on all the cluster nodes.

Run `swinstall(1m)` to install ContinentalClusters product from an SD depot. The toolkit control file and contributed scripts will reside in the `/opt/cmcluster/toolkit/SGCA` directory. One fileset contains the Continuous Access (CA) template control file and sample scripts directory.

3. When `swinstall(1m)` has completed, create a directory as follows for the new package in the recovery cluster:

   ```
   # mkdir /etc/cmcluster/<package_name>
   ```

Create an MC/ServiceGuard package configuration file in the primary cluster with the commands:
Physical Data Replication for ContinentalClusters Using Continuous Access XP

Setting up a Recovery Package on the Recovery Cluster

# cd /etc/cmcluster/<package_name>

# cmmakepkg -p <package_name>.ascii

Customize it as appropriate to your application. Be sure to include the pathname of the control script (/etc/cmcluster/<package_name>/<package_name>.cntl) for the RUN_SCRIPT and HALT_SCRIPT parameters.

Set the AUTO_RUN flag (PKG_SWITCHING_ENABLED in ServiceGuard A.11.09) to NO. This is to ensure the package will not start when the cluster starts. Do not use cmmodpkg to enable package switching on any recovery package. Enabling package switching will automatically start the recovery package. Package switching on a recovery package will be automatically set by the cmrecovercl command on the recovery cluster when it successfully starts the recovery package.

4. Copy and rename the Continuous Access XP package control script from a system in the primary cluster to this new directory. Note that each package has its own directory and its own uniquely named control script.

**NOTE**

Do not create an MC/ServiceGuard package control script template with the cmmakepkg -s command or by using SAM. The package control script template included with this product should be used instead.

The scripts in the Samples-CC directory are examples of ways to automate some of the tasks. These scripts are for your convenience and are not supported. These example scripts should be copied to another directory (e.g., /etc/cmcluster/SGCA) and then edited to customize them for the specific environment.

5. Edit the package control script <bkpackage_name>.cntl as follows:

   a. Change the PKGDIR variable and set it to the full path name of the directory where the control script has been placed. This directory must be unique for each package and is used for status data files.

   b. Uncomment the DC2HOST array variable and set the elements to the hostnames of the recovery cluster. The order of the hostnames is not important.

   c. Make sure that the behavioral configuration environment variables AUTO_* are uncommented. It is recommended that you
Setting up a Recovery Package on the Recovery Cluster

retain the default values of these variables unless you have a specific business requirement to change them. See the template for an explanation of these variables.

6. Edit the remaining control script variables (IP, SUBNET, SERVICE_NAME, SERVICE_CMD and SERVICE_RESTART), on the recovery cluster, according to the needs of the application as it runs on the recovery cluster. See the MC/ServiceGuard manual for more information on these variables.

---

**NOTE**

Some of the control script variables, such as VG and LV, on the recovery cluster must be the same as on the primary cluster. Some of the control script variables, such as, FS, SERVICE_NAME, SERVICE_CMD and SERVICE_RESTART are probably the same as on the primary cluster. Some of the control script variables, such as IP and SUBNET, on the recovery cluster are probably different from those on the primary cluster. Make sure that you review all the variables accordingly.

---

7. If the customer-defined run and halt commands are different on the recovery cluster, edit the appropriate places according to the needs of the application. See the MC/ServiceGuard manual for more information on these functions.

8. Make sure the CLUSTER_TYPE variable is set to continental.

9. Distribute Continuous Access XP package control script files to other nodes in the recovery cluster by using `ftp` or `rcp`:

   ```
   # rcp -p /etc/cmcluster/<bkpkg_name>/<bkpkg_name>.cntl \
   other_node:/etc/cmcluster/<bkpkg_name>/<bkpkg_name>.cntl
   ```

   When using `ftp`, be sure to make the file executable on any destination systems.

10. Apply the MC/ServiceGuard configuration using the `cmapplyconf` command or SAM.

11. Verify that each host in the continental cluster has the following files in the directory `/etc/cmcluster/<bkpackage_name>`:

    - `<bkpkg_name>.cntl`—Continuous Access XP package control script
    - `<bkpkg_name>.ascii`—MC/ServiceGuard package ASCII
Physical Data Replication for ContinentalClusters Using Continuous Access XP

Setting up a Recovery Package on the Recovery Cluster

configuration file

- `<bkpkg_name>.sh`—Package monitor shell script, if applicable

12. Edit the file `/etc/rc.config.d/raidmgr`, specifying the Raid Manager instance to be used for ContinentalClusters, and specify that the instance be started at boot time.

---

**NOTE**

The appropriate Raid Manager instance used by ContinentalClusters must be running before the package is started. This normally means that the Raid Manager instance must be started before MC/ServiceGuard is started.

---

13. Make sure the packages on the primary cluster are not running. Using standard MC/ServiceGuard commands (`cmruncl`, `cmhaltcl`, `cmrunpkg`, `cmhaltpkg`) test the recovery cluster for cluster and package startup and package failover.

14. Any running package on the recovery cluster that has a counterpart on the primary cluster should be halted at this time.
Setting up the Continental Cluster Configuration

The steps below are the basic procedure for setting up the ContinentalClusters configuration file and the monitoring packages on the two clusters. For complete details on creating and editing the configuration file, refer to Chapter 5, “Building a Continental Cluster.”

1. Generate the ContinentalClusters configuration using the following command:

   ```
   # cmqueryconcl -C cmconcl.config
   ```

2. Edit the configuration file cmconcl.config with the names of the two clusters, the nodes in each cluster, the recovery groups and the monitoring definitions. The recovery groups define the primary and recovery packages. When data replication is done using Continuous Access XP, there are no data sender and receiver packages.

   Define the monitoring parameters, the notification mechanism (ITO, email, console, SNMP, syslog or tcp) and notification type (alert or alarm) based on the cluster status (unknown, down, up or error). Descriptions for these can be found in the configuration file generated in the previous step.

3. Edit the continental cluster security file `/etc/opt/cmom/cmomhosts` to allow or deny hosts read access by the monitor software.

4. On all nodes in both clusters copy the monitor package files from `/opt/cmcond/scripts` to `/etc/cmcluster/ccmonpkg`. Edit the monitor package configuration as needed in the file `/etc/cmcluster/ccmonpkg/ccmonpkg.config`. Set the `AUTO_RUN` flag to `YES`. This is in contrast to the flag setting for the application packages. We want the monitor package to start automatically when the cluster is formed.

5. Apply the monitor package to both cluster configurations using the following command:

   ```
   # cmapplyconf -P /etc/cmcluster/ccmonpkg/ccmonpkg.config
   ```

6. Start the monitor package on both clusters.
Setting up the Continental Cluster Configuration

NOTE
The monitor package for a cluster checks the status of the other cluster and issues alerts and alarms, as defined in the ContinentalClusters configuration file, based on the other cluster’s status.

7. Check /var/adm/syslog/syslog.log for messages and check for running processes cmclrmond and /opt/cmconcl/jre/bin/PA_RISC/native_threads/jre. Also check the ccmonpkg package log file.

8. Apply the continental cluster configuration file using cmapplyconcl. Files are placed in /var/adm/cmconcl/instances. There is no change to /etc/cmcluster/cmclconfig nor is there an equivalent file for ContinentalClusters. Example:

   # cmapplyconcl -C cmconcl.config

NOTE
You must create or edit /.rhosts on all the systems in the continental cluster to allow write access to the system running the cmapplyconcl command. After the ContinentalClusters configuration file is successfully generated the file or system entries can be deleted.

9. Start the primary packages on the primary cluster using cmrunpkg. Test local failover within the primary cluster.

10. View the status of the ContinentalCluster primary and recovery clusters, including configured event data:

   # cmviewconcl -v

   The continental cluster is now ready for testing. See “Testing the Continental Cluster” on page 182.
Switching to the Recovery Cluster in Case of Disaster

It is vital the administrator verify that recovery is needed after receiving a cluster alert or alarm. Network failures may produce false alarms. After validating a failure, start the recovery process using the `cmrecovercl [-f]` command. Note the following:

- During an alert, the `cmrecovercl` will not start the recovery packages unless the `-f` option is used.
- During an alarm, the `cmrecovercl` will start the recovery packages without the `-f` option.
- When there is neither an alert nor an alarm condition, `cmrecovercl` cannot start the recovery packages on the recovery cluster. This condition applies not only when no alert or alarm was issued, but also applies to the situation where there was an alert or alarm, but the primary cluster recovered and its current status is Up.
Failback Scenarios

The goal of HP ContinentalClusters is to maximize system and application availability. However, even systems configured with ContinentalClusters can experience hardware failures at the primary site or the recovery site, as well as the hardware or networking failures connecting the two sites. The following discussion addresses some of those failures and suggests recovery approaches applicable to environments using data replication provided by HP Surestore E XP series disk arrays and Continuous Access (CA). In Chapter 5, “Building a Continental Cluster,” there is a discussion of failback mechanisms and methodologies in “Restoring Disaster Tolerance” on page 190.

Scenario 1

The primary site has lost power, including backup power (UPS), to both the systems and disk arrays that make up the MC/ServiceGuard Cluster at the primary site. There is no loss of data on either the XP disk array or the operating systems of the systems at the primary site.

Scenario 2

The primary site XP disk array experienced a catastrophic hardware failure and all data was lost on the array.

Failback in Scenarios 1 and 2

After reception of the ContinentalClusters alerts and alarm, the administrators at the recovery site follow the prescribed processes and recovery procedures to start the protected applications on the recovery cluster. Each ContinentalClusters package control script will evaluate the status of the XP paired volumes. Since neither the systems nor the XP disk array at the primary site are accessible, the control file will initially report the paired volumes with a local status of SVOL_PAIR or SVOL_PSUE (in ASYNC mode) and a remote status of EX_ENORMT, PSUE or PSUS, indicating that there is an error accessing the primary site. The control file script is programmed to handle this condition and will enable the volume groups, mount the logical volumes, assign floating IP addresses and start any processes as coded into the script.
Physical Data Replication for ContinentalClusters Using Continuous Access XP
Failback Scenarios

NOTE

In **ASYNC** mode, the package will halt unless a force flag is present or unless the auto variable **AUTO_SVOLPSUE** is set to 1.

The fence level of the paired volume—**NEVER**, **ASYNC**, or **DATA**—will not impact the starting of the packages at the recovery site. The control script will perform the following command with regards to the paired volume:

```
# horctakeover -g <dev-grp-name> -S
```

Subsequently, the paired volume will have a status of **SVOL_SWSS**. To view the local status of the paired volumes run:

```
# pairvolchk -g <dev-grp-name> -s
```

To view the remote status of the paired volumes, run:

```
# pairvolchk -g <dev-grp-name> -c
```

(While the remote XP disk array and primary cluster systems are down, the command will time out with an error code of 242.)

After power is restored to the primary site, or when a newly configured array is brought online, the XP paired volumes may have either a status of **PVOL_PSUE** on the primary site or **SVOL_SWSS** on the secondary site. The following procedure applies to this situation:

1. While the package is still running, issue the following command from the recovery host:

   ```
   # pairresync -g <dgname> -c 15 -swaps
   ```

   This starts the resynchronization, which can take a long time if the entire primary disk array was lost or a short time if the primary array was intact at the time of failover.

2. When resynchronization is complete, halt the ContinentalClusters recovery packages at the recovery site using the command

   ```
   # cmhaltpkg <pkg_name>
   ```

   This will halt any applications, remove any floating IP addresses, unmount file systems and deactivate volume groups as programmed into the package control files. The status of the paired volumes will remain **SVOL_PAIR** at the recovery site and **PVOL_PAIR** at the primary site.
Failback Scenarios

3. Start the cluster at the primary site. Assuming they have been properly configured, the ContinentalClusters primary packages should not start. The monitor package should start automatically.

4. Manually start the ContinentalClusters primary packages at the primary site using the command
   
   ```
   # cmrunpkg <pkg_name>
   ```

   The control script is programmed to handle this case.

5. Ensure that the monitor packages at the primary and recovery sites are running.

Scenario 3

The primary site has lost power, including backup power (UPS), to both the systems and disk arrays that make up the MC/ServiceGuard Cluster at the primary site. There is no loss of data on either the XP disk array or the operating systems of the hosts at the primary site. The system administrator desires to switch the identities of the two sites (i.e., the old or original primary site will become the recovery site and the old or original recovery site will become the primary site).

Failback in Scenario 3

After reception of the ContinentalClusters alerts and alarm, the administrators at the recovery site follow the prescribed processes and recovery procedures to start the protected applications on the recovery cluster. Each ContinentalClusters package control script will evaluate the status of the XP paired volumes. Since neither the systems nor the XP disk array at the primary site are accessible, the control file will initially report the paired volumes with a local status of `SVOL_PAIR` or `SVOL_PSUE` and a remote status of `EX_ENORMT`, `PSUE` or `PSUS`, indicating that there is an error accessing the primary site. The control file script is programmed to handle this condition and will enable the volume groups, mount the logical volumes, assign floating IP addresses and start any processes as coded into the script. The fence level of the paired volume, either `NEVER`, `ASYNC`, or `DATA`, will not impact the starting of the packages at the recovery site. The control script will perform the following command with regards to the paired volume:

   ```
   # horctakeover -g <dev-grp-name> -S
   ```

   Subsequently, the paired volume will have a status of `SVOL_SWSS`. To
Physical Data Replication for ContinentalClusters Using Continuous Access XP
Failback Scenarios

view the local status of the paired volumes run
# pairvolchk -g <dev-grp-name> -s

To view the remote status of the paired volumes run
# pairvolchk -g <dev-grp-name> -c

(While the remote XP disk array and primary cluster systems are down, the command will timeout with an error code of 242).

After power is restored to the primary site, the XP paired volumes may have either a status of PSUE or SMPL. The procedure to switch the functionality of the primary and recovery sites (i.e., the primary site becomes the recovery site and the recovery site becomes the primary site) is substantially the same in either case.

1. While the recovery packages are running, for each paired volume with a status of PSUE, split the paired volume at the old recovery site using

   # pairsplit -g <dev-grp-name> -S

   If the primary site paired volume status is currently SMPL proceed to the next step.

2. For each paired volume manually recreate the volume using

   # paircreate -g <dev-grp-name> -f <fence-level> -vl -c 15

   synchronizing from the local (option -vl) side. Fence level may be DATA, ASYNC, or NEVER.

3. Halt the monitor package at the recovery site. The recovery packages should be running at this time.

4. Halt the monitor package at the primary site.

5. Reconfigure the ContinentalClusters configuration file. Section 1 is not changed, unless systems are added, deleted or their names changed or a cluster name is changed. In Section 2, reverse the designation for the primary and recovery packages. Modify Section 3 only if different monitoring was defined for the original primary and recovery sites. If the monitoring was equivalent no changes are necessary.

6. Apply the new ContinentalClusters configuration file using:

   # cmapplyconcl -C <CC-config-file>

7. Start the monitor packages at both the primary and recovery sites.
Physical Data Replication for ContinentalClusters Using Continuous Access XP

Failback Scenarios

8. View the status of the ContinentalCluster primary and recovery clusters, including configured event data:

   # cmviewconcl -v

**Failback When the Primary Has SMPL Status**

The following procedure applies to the situation where the primary site paired volumes have a status that has been set to SMPL, possibly through manual intervention:

1. Halt the ContinentalClusters recovery packages at the recovery site using the command

   # cmhaltpkg <pkg_name>

   This will halt any applications, remove any floating IP addresses, unmount file systems and deactivate volume groups as programmed into the package control files. The status of the paired volumes will remain SMPL at the recovery site and PSUE at the primary site.

2. Start the cluster at the primary site. Assuming they have been properly configured the ContinentalClusters primary packages should not start. The monitor package should start automatically.

3. Since the paired volumes have a status of SMPL at both the primary and recovery sites, the XP views the two halves as unmirrored. From a system at the primary site, manually create the paired volume:

   # paircreate -g <dev-grp-name> -f <fence-level> -vr -c 15

   Since the most current data will be at the remote or recovery site, this will synchronize the data from the remote or recovery site (use of the -vr option directs the command to synchronize from the remote site). Wait for the synchronization process to complete before proceeding to the next step. Failure to wait for the synchronization to complete will result in the package failing to start in the next step.

4. Manually start the ContinentalClusters primary packages at the primary site using the command

   # cmrunpkg <pkg_name>

   The control script is programmed to handle this case. The control script recognizes that the paired volume is synchronized and will proceed with the programmed package startup.

5. Ensure that monitor packages are running at both sites.
Maintaining the Continuous Access XP Data Replication Environment

There might be situations where a package has to be taken down for maintenance purposes without having the package move to another node. The following procedure is recommended for normal maintenance of ContinentalClusters:

1. Shut down the package with the appropriate MC/ServiceGuard command. Example:
   ```
   # cmhaltpkg <pkg_name>
   ```

2. Distribute the ContinentalClusters configuration changes. On the primary cluster:
   ```
   # cmapplyconf -P <pkgconfig>
   ```
   On the recovery cluster:
   ```
   # cmapplyconf -P <bkpkgconfig>
   ```

3. Start up the package with the appropriate MC/ServiceGuard command. Example:
   ```
   # cmmodpkg -e <pkg_name>
   ```

   **NOTE**

   Never enable package switching on both the primary package and the recovery package.

4. To apply a new ContinentalClusters configuration shut down any monitor packages (`ccmonpkg`), run `cmapplyconcl` and then restart the monitor packages.

**Rescynchronizing**

After certain failures, data are no longer remotely protected. In order to restore disaster-tolerant data protection after repairing or recovering from the failure, you must manually run the command `pairresync`. This command must successfully complete for disaster-tolerant data protection to be restored. Following is a partial list of failures that
Physical Data Replication for ContinentalClusters Using Continuous Access XP

Maintaining the Continuous Access XP Data Replication Environment

require running pairresync to restore disaster-tolerant data protection:

- failure of ALL CA (ESCON) links without restart of the application
- failure of ALL CA links with Fence Level DATA with restart of the application on a primary host
- failure of the entire recovery Data Center for a given application package
- failure of the recovery XP disk array for a given application package while the application is running on a primary host

Following is a partial list of failures that require full resynchronization to restore disaster-tolerant data protection. Full resynchronization is automatically initiated by moving the application package back to its primary host after repairing the failure.

- failure of the entire primary Data Center for a given application package
- failure of all of the primary hosts for a given application package
- failure of the primary XP disk array for a given application package
- failure of all CA links with application restart on a secondary host

NOTE

The preceding steps are automated provided the default value of 1 is being used for the auto variable AUTO_PSUESUS. Once the ESCON link failure has been fixed, the user only needs to halt the package at the failover site and restart on the primary site. However, if you want to reduce the amount of application downtime, you should manually invoke pairresync before failback.

Full resynchronization must be manually initiated as described in the next section) after repairing the following failures:

- failure of the recovery XP disk array for a given application package followed by application startup on a primary host
- failure of all CA links with Fence Level NEVER or ASYNC with restart of the application on a primary host

Pairs must be manually recreated if both the primary and recovery XP disk arrays are in the SMPL (simplex) state.
Physical Data Replication for ContinentalClusters Using Continuous Access XP

Maintaining the Continuous Access XP Data Replication Environment

Make sure you periodically review the following files for messages, warnings and recommended actions. You should particularly review these files after system, data center and/or application failures:

- `/var/adm/syslog/syslog.log`
- `/etc/cmcluster/<package-name>/<package-name>.log`
- `/etc/cmcluster/<bkpackage-name/<bkpackage-name>.log`

Using the pairresync Command

The pairresync command can be used with special options after a failover in which the recovery site has started the application and has processed transaction data on the disk at the recovery site, but the disks on the primary site are intact. After the ESCON link is fixed, you use the pairresync command in one of the following two ways depending on which site you are on:

- `pairresync -swapp` — from the primary site.
- `pairresync -swaps` — from the failover site.

These options take advantage of the fact that the recovery site maintains a bit-map of the modified data sectors on the recovery array. Either version of the command will swap the personalities of the volumes, with the PVOL becoming the SVOL and SVOL becoming the PVOL. With the personalities swapped, any data that has been written to the volume on the failover site (now PVOL) are then copied back to the SVOL now running on the primary site. During this time the package continues running on the failover site. After resynchronization is complete, you can halt the package on the failover site, and restart it on the primary site. MetroCluster will then swap the personalities between the PVOL and the SVOL, returning PVOL status to the primary site.

Some Further Points

- This toolkit may increase package startup time by 5 minutes or more. Packages with many disk devices will take longer to start up than those with fewer devices due to the time needed to get device status from the XP disk array or to synchronize.

NOTE

Long delays in package startup time will occur in those situations when
Physical Data Replication for ContinentalClusters Using Continuous Access XP

Maintaining the Continuous Access XP Data Replication Environment

recovering from broken pair affinity. For example, restarting the application on a primary host after it has been running on a recovery host (with the SVOL changed to SMPL), results in a paircreate operation. Any paircreate operation performs a full copy of the volumes which can take from minutes to hours depending on the amount of data being copied.

- The value of RUN_SCRIPT_TIMEOUT in the package ASCII file should be set to NO_TIMEOUT or to a large enough value to take into consideration the extra startup time due to getting status from the XP disk array. See the previous paragraph for more information on the extra startup time.

- Online cluster configuration changes may require a Raid Manager configuration file to be changed. Whenever the configuration file is changed, the Raid Manager instance must be stopped and restarted. The Raid Manager instance must be running before any ContinentalClusters package movement occurs.

- A given file system must not reside on more than one XP frame for either the PVOL or the SVOL. A given LVM Logical Volume (LV) must not reside on more than one XP frame for either the PVOL or the SVOL.

- The application is responsible for data integrity, and must use the O_SYNC flag when ordering of I/Os is important. Most relational database products are examples of applications that ensure data integrity by using the O_SYNC flag.

- Each host must be connected to only the XP disk array that contains either the PVOL or the SVOL. A given host must not be connected to both the PVOL and the SVOL of a continuous access pair.
Physical Data Replication for ContinentalClusters Using EMC SRDF

This chapter shows how to use the MetroCluster/SRDF package control script template to set up data replication between the primary and secondary packages in a Continental Cluster that uses the EMC Symmetrix Disk Array with the Symmetrix Remote Data Facility (SRDF). These packages run on different clusters within the Continental cluster configuration. It is assumed that readers are already familiar with MC/ServiceGuard configuration tasks, MC/ServiceGuard Continental Cluster concepts and installation procedures, EMC Symmetrix and SRDF concepts and SymCLI configuration and use.

This chapter has the following sections:

• Overview
• Preparing a Continental Cluster for Data Replication
• Creating and Exporting Volume Groups
• Setting up a Primary Package on the Primary Cluster
• Setting up a Recovery Package on the Recovery Cluster
• Setting up the Continental Cluster Configuration
• Switching to the Recovery Cluster in Case of Disaster
• Failback Scenarios
• Maintaining the EMC SRDF Data Replication Environment

NOTE

This chapter is for ContinentalClusters users only. Users of MetroCluster with EMC SRDF should not use the procedures in this chapter. Refer instead to Chapter 4, “Building a Metropolitan Cluster Using MetroCluster/SRDF.”
Overview

Physical data replication with the EMC Symmetrix Disk Array uses the EMC SRDF package control script template, which is installed in the /opt/cmcluster/toolkit/SGSRDF directory when you install ContinentalClusters. This directory includes the following files:

- **sgsrdfpkg.cntl**—the EMC SRDF package control script template. This template must be customized for the specific EMC Symmetrix and HP 9000 host system configuration. Copies of this template must be customized for each separate MC/ServiceGuard package and for each MC/ServiceGuard Cluster.

- **Samples-CC**—a directory containing sample convenience shell scripts that must be edited before using. These shell scripts may help to automate some configuration tasks. These scripts are contributed, and not supported. This directory also contains example Raid Manager configuration files and other examples.

The EMC SRDF package control script template should be used on all continental cluster nodes and clusters where the following are true:

- The application will run as a primary package on the primary cluster and, following failover, as a recovery package in the recovery cluster.
- Data are on an EMC Symmetrix ICDA connected to the primary cluster and replicated with SRDF to a second EMC Symmetrix ICDA connected to the recovery cluster.

To use EMC SRDF data replication in a continental cluster, the following must also be installed and configured on each HP 9000 host system that might execute the application:

- HP-UX 11.0
- Symmetrix Command Line Interface (SymCLI) software
- Pass-through SCSI driver subsystem (included with HP-UX)
- MC/ServiceGuard A.11.09 or greater

The procedures below assume that you are configuring one or more MC/ServiceGuard application packages whose data reside on the EMC disks and that data is replicated between the two EMC disk arrays with SRDF. The procedure must be repeated for each application package.
Preparing a Continental Cluster for Data Replication

These procedures will prepare your MC/ServiceGuard clusters for use in a ContinentalClusters configuration.

1. Ensure that the Symmetrix ICDAs are correctly cabled to each host system in the cluster that will run packages whose data reside on the Symmetrix.

   **NOTE**
   Fibre Channel host adapters with the Fibre Channel/SCSI Mux, native Fibre Channel host adapters (connected directly, through hubs or through switches) to the EMC SRDF or F/W SCSI host adapters directly connected to the EMC SRDF may be used.

2. Install the SymCLI software on each host system in both clusters that has data residing on the Symmetrix. The default directory is /usr/symcli/bin.

3. Create the SymCLI database on each system according to the instructions in the SymCLI manual. Run `symcfg discover` to create the Symmetrix database on all nodes in both clusters.

   ```bash
   # symcfg discover
   ```

   **NOTE**
   Make sure that you do not set the SYM_SID and SYM_DG environment variables before running the symcfg command.

4. Get a list of data for the Symmetrix devices available. Use the `syminq` command (without any switches) to get a list:

   ```bash
   #syminq
   ```

   The list includes the following data for each Symmetrix logical device:
   - HP-UX special device filename
   - device type (R1, R2, BCV or blank)
Preparing a Continental Cluster for Data Replication

- Symmetrix serial number
  Gatekeeper devices are usually of size 2880. The Symmetrix device serial number is useful in matching the devices to the actual devices in the Symmetrix configuration (downloaded by EMC). The following is a key to interpreting the number:

  - The first two digits equal the last two digits of the serial number of the Symmetrix frame.
  - The next three hexadecimal digits equal the Symmetrix device number that is seen in the output of the status command:
    
    ```sh
    # symrdf -g <symdevgrpname> query
    ```

    and is used by the ContinentalCluster/SRDF control script and saved in the file

    ```
    /etc/cmcluster/<pkg_name>/symrdf.out
    ```

    The contents of this file may be useful for debugging purposes.

  - The next three digits equal the Symmetrix host adapter (SA or FA) and port numbers; this is useful to see multiple host links to the same Symmetrix device (PVLinks).

5. Get a list of the disk devices known to the host system. Use the following command:

   ```sh
   # insf -e
   ```

   Then run one of the following:

   ```sh
   # ioscan -kfnC disk
   ```

   or the preferred form:

   ```sh
   # ioscan -fnC disk
   ```

   Match the device file names in this listing with the device file names in the output from the `syminq` command to see which devices are seen from this host to make sure that this host can see all necessary devices.

6. Use the Symmetrix ID to determine to which Symmetrix the host is connected. Then use the Symmetrix device number to determine which devices are the same logical device seen by each host system that is connected to the same Symmetrix unit and record the special device file names for later use.

   It is recommended that you create a table with:
Preparing a Continental Cluster for Data Replication

- hostname
- Symmetrix ID (first two digits of the device serial number)
- Symmetrix device number (next three hexadecimal digits of the device serial number)
- Symmetrix device type (BCV, R1, R2, or gatekeeper)

to help you correlate the devices on all of the host systems. The table might look like Table 7-1.

**Table 7-1** Symmetrix Device and HP-UX Device Correlation

<table>
<thead>
<tr>
<th>Symmetrix ID, device #, and type</th>
<th>Node 1 /dev/rdsk device file name</th>
<th>Node 2 /dev/rdsk device file name</th>
<th>Node 3 /dev/rdsk device file name</th>
<th>Nodes 4 /dev/rdsk device file name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID 95, Dev# 005, Type R1</td>
<td>c0t4d0</td>
<td>c6t0d0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ID 50, Dev# 014, Type R2</td>
<td></td>
<td>c4t0d0</td>
<td></td>
<td>c0t4d0</td>
</tr>
<tr>
<td>ID 95, Dev# 00A, Type R2</td>
<td></td>
<td></td>
<td></td>
<td>c0t4d0</td>
</tr>
<tr>
<td>ID 50, Dev# 012, Type R1</td>
<td></td>
<td></td>
<td></td>
<td>c4t3d2</td>
</tr>
<tr>
<td>ID 95, Dev# 040, Type GK</td>
<td>c0t15d0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ID 50</td>
<td></td>
<td></td>
<td></td>
<td>c3t15d1</td>
</tr>
</tbody>
</table>
Preparing a Continental Cluster for Data Replication

In this example, the target and LUN numbers do not always match for all hosts on the same side of the SRDF link. For ease of administration, it is recommended that all hosts use the same target and LUN number when referring to the same Symmetrix logical device.

You will probably need to refer to the actual Symmetrix configuration (available from the Symmetrix console or by using the Symmetrix Manager for Open Systems) to definitively match up the R1 devices in one Symmetrix unit with the R2 devices in the other Symmetrix unit.

This table will be useful when you complete Step 8 below. It will allow you to easily (and correctly) specify the device file names when creating the Symmetrix device groups.

Note that the Symmetrix device number may be the same or different each of the Symmetrix units for the same logical device. In other words, the device number for the logical device on the R1 side of the SRDF link may be different from the device ID for the logical device on the R2 side of the SRDF link.

When you are determining the configuration for the Symmetrix devices for a new installation, it is recommended that you try to use the same Symmetrix device number for the R1 devices and the R2 devices. It is also recommended that the same target and LUN number be configured for all hosts that have access to the same Symmetrix logical device.

7. Use the symdg, symld and symgate commands on all nodes in both clusters to create the Symmetrix device groups. All devices

Table 7-1 Symmetrix Device and HP-UX Device Correlation

<table>
<thead>
<tr>
<th>Symmetrix ID, device #, and type</th>
<th>Node 1 /dev/rdsk device file name</th>
<th>Node 2 /dev/rdsk device file name</th>
<th>Node 3 /dev/rdsk device file name</th>
<th>Nodes 4 /dev/rdsk device file name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dev# 041</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type GK</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ID 95</td>
<td>c1t1d0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dev# 028</td>
<td></td>
<td></td>
<td></td>
<td>c6t1d0</td>
</tr>
<tr>
<td>Type BCV</td>
<td></td>
<td></td>
<td>n/a</td>
<td>n/a</td>
</tr>
</tbody>
</table>

In this example, the target and LUN numbers do not always match for all hosts on the same side of the SRDF link. For ease of administration, it is recommended that all hosts use the same target and LUN number when referring to the same Symmetrix logical device.
Preventing a Continental Cluster for Data Replication

belonging to volume groups that are owned by an application package must be defined in a single Symmetrix device group. The device group may be different between the R1 side and the R2 side of the SRDF link. A unique Symmetrix gatekeeper device should be created for each application package.

When creating the Symmetrix device groups, you may specify only one HP-UX path to a particular Symmetrix device. Do not specify alternate paths (PVLinks). The SymCLI uses the HP-UX path only to determine to which Symmetrix device you are referring. The Symmetrix device may be added to the device group only once.

Use only the default Symmetrix Logical Device names of the form "DEVnnn" (e.g., DEV001). Do not use the option for creating your own device names.

See the scripts in the Samples-CC directory for examples of how you would use these commands to build the SymCLI database and setup the Symmetrix device groups. The sample scripts are:

- mk3symgrps.<hostname>
- mk4gatekpr.<hostname>

The script must be customized for each system in both clusters. Customizations include:

- particular special device file names
- Symmetrix device group name (arbitrary, but unique name may be chosen for each group that defines all of the volume groups (VGs) belonging to a particular MC/ServiceGuard package)
- keyword RDF1 or RDF2

Gatekeeper devices must be unique per MC/ServiceGuard package to prevent contention in the Symmetrix when commands are issued. Gatekeeper devices are unique to a Symmetrix unit. They are not replicated across the SRDF link.

There should be at least two gatekeeper devices accessible from each host system. Each gatekeeper device is configured on different physical links. The SymCLI will switch to an alternate gatekeeper device if the path to the primary gatekeeper device fails.

You can display what is in the SymCLI database with the commands:

- symdg list
Preparing a Continental Cluster for Data Replication

- `symld -g <symdevgrpname> list`
- `symgate list`

You may also use the following command to get a listing of all devices and their states:

```
# symrdf list
```
Creating and Exporting Volume Groups

Create and export the volume groups using the following procedure:

1. Define the appropriate volume groups on each host system that might run the application package. Use the commands:

   # mkdir /dev/vgxx
   # mknod /dev/vgxx/group c 64 0xnn0000

   where the name /dev/vgxx and the number nn are unique within the entire cluster.

2. Create the volume group only on the primary system. Use the `vgcreate` and perhaps the `vgextend` command, specifying the appropriate special device file names.

3. Export the volume groups on the primary system without removing the special device files. Use the commands:

   # vgchange -a n <vgname>
   # vgexport -s -p -m <mapfilename> <vgname>

   Note: The `vgexport -s` option creates the special device files for the `vgimport` on adoptive nodes. The `vgimport -s` option will look for the marked disks for that volume group. It is expedient, and reduces error. Unfortunately the alternate pvlink optimization may be sacrificed. For lots of disks it is a trade off.

   Make sure that you copy the mapfiles to all of the host systems.

   NOTE

   See the example scripts `add_vgpkgCCA.sh` and `vgexp_vgpkgCCA.sh` in the `Samples-CC` directory to see how to automate steps one through three.

4. Import the VGs on all of the other systems that might run the MC/ServiceGuard package and backup the LVM configuration. Do this on both the primary and recovery cluster nodes. Make sure that you split the logical SRDF links before importing the VGs. A sample script to split the links is `pre.cmquery`.

   Use the commands:
Physical Data Replication for ContinentalClusters Using EMC SRDF

Creating and Exporting Volume Groups

```
# symrdf -g <symdevgrpname> split -v
# vgimport -s -m <mapfilename> <vgname>
# vgchange -a y <vgname>
# vgcfgbackup <vgname>
# vgchange -a n <vgname>
# symrdf -g <symdevgrpname> establish -v
# symrdf list
```

**NOTE**

The `vgexport -s` option creates the special device files for the `vgimport` on adoptive nodes. The `vgimport -s` option will look for the marked disks for that volume group. It is expedient, and reduces error. Unfortunately the alternate pvlink disk performance may be sacrificed. For lots of disks in the configuration, while simpler, it is a trade off.

See the sample script `Samples-CC/vgimp_vgpkgCCA.sh`.

5. Before MC/ServiceGuard and ContinentalClusters can be configured, HP-UX must be satisfied. Confirm the disks have shared disk, and the VG can be seen on the primary and adoptive nodes. Use the following test procedures:

   a. Confirm the volume groups have been imported using command
      ```
      # strings /etc/lvmtab
      ```

   b. Split the SRDF links (if linked) using:
      ```
      # Samples-CC/pre.cmquery
      ```

   c. Activate the volume group on the primary Node in the Primary Cluster, and mount LV’s. Do they appear? If so, then deactivate the VG as follows:
      ```
      # vgchange -a y vgpkgCC
      # mount /dev/vgpkgCCA/lvpkgCCA_01 /pkgCCA_01
      # mount /dev/vgpkgCCA/lvpkgCCA_02 /pkgCCA_02
      # bdf
      ```
      If the mounted volumes show, then:
      ```
      # vgchange -a n vgpkgCCA
      ```
d. Activate the volume group on all failover nodes in the primary cluster. On each node, activate the volume group, and mount LV’s. Do they appear? If so, then deactivate the VG as follows:

```
# vgchange -a y vgpkgCCA
# mount /dev/vgpkgCCA/lvpkgCCA_01 /pkgCCA_01
# mount /dev/vgpkgCCA/lvpkgCCA_02 /pkgCCA_02
# bdf
```

If the mounted volumes show, then:
```
# vgchange -a n vgpkgCCA
```

e. Keep the SRDF links split until told to relink.
Setting up a Primary Package on the Primary Cluster

Use the procedures in this section to configure a primary package on the primary cluster. Consult the MC/ServiceGuard documentation for more detailed instructions on setting up MC/ServiceGuard with packages, and for instructions on how to start, halt, and move packages and their services between nodes in a cluster.

1. Create and test a standard MC/ServiceGuard cluster using the procedures described in the user’s manual, Managing MC/ServiceGuard.

2. If this was not done previously, split the EMC SRDF logical links for the disks associated with the application package. See the script Samples-CC/pre.cmquery (edit to the SRDF groups configured) for an example of how to automate this task. The script must be customized with the Symmetrix device group names.

3. Install ContinentalClusters on all the cluster nodes in the primary cluster (Skip this step if the software has been preinstalled)

**NOTE**
MC/ServiceGuard should already be installed on all the cluster nodes.

Run swinstall(1m) to install ContinentalClusters product from an SD depot. The toolkit control file and contributed scripts will reside in the /opt/cmcluster/toolkit/SGSRDF directory. One fileset contains the EMC SRDF template control file and sample scripts directory.

4. When swinstall(1m) has completed, create a directory as follows for the new package in the primary cluster:

   ```bash
   # mkdir /etc/cmcluster/<pkg_name>
   ```

5. Copy the EMC SRDF package control script to this new directory and rename it to <pkg_name>.cntl. Note that each package has its own directory and its own uniquely named control script.

**NOTE**
Do not create an MC/ServiceGuard package control script template with
Physical Data Replication for ContinentalClusters Using EMC SRDF

Setting up a Primary Package on the Primary Cluster

The cmmakepkg -s command or by using SAM. The package control script template included with this product should be used instead. Copy the primary package control file from the sample script found in /opt/cmcluster/toolkit/SGSRDF/srdfpkg.cntl.

The scripts in the Samples-CC directory are examples of ways to automate some of the tasks. These scripts are for your convenience and are not supported. These example scripts should be copied to another directory (e.g., /etc/cmcluster/SGSRDF) and then edited to customize them for the specific environment.

6. Create an MC/ServiceGuard Application package configuration file with the commands:

```bash
# cd /etc/cmcluster/<pkg_name>
# cmmakepkg -p <pkg_name>.conf
```

Customize it as appropriate to your application. Be sure to include Node names, the pathname of the control script (/etc/cmcluster/<pkg_name>/<pkg_name>.cntl) for the RUN_SCRIPT and HALT_SCRIPT parameters.

Also change AUTO_RUN (PKG_SWITCHING_ENABLED in ServiceGuard A.11.09) to NO. This will ensure that the application packages will not start automatically. (the cmonpkg will be set to yes) Define the service (as required)

7. Edit the package control script <pkg_name>.cntl as follows:

Add the path where the SymCLI software binaries have been installed to the PATH environment variable. The usual location is currently /usr/symcli/bin

Uncomment AUTOR1RWNL, AUTOR1UIP, AUTOR2RWNL and AUTOR2WDNL environment variables. It is recommended that you retain the default values of these variables unless you have a specific business requirement to change them. See the template for an explanation of these variables.

a. Uncomment the PKGDIR variable and set it to the full path name of the directory where the control script has been placed. This directory must be unique for each package and is used for status data files. For example, set PKGDIR to /etc/cmcluster/<pkg_name>.
Physical Data Replication for Continental Clusters Using EMC SRDF

Setting up a Primary Package on the Primary Cluster

b. Uncomment the `PKGR1SYMGRP` and `PKGR2SYMGRP` variables and set them to the Symmetrix device group names given in the ‘symdg’ command.

c. Uncomment the `R1HOST` array variable only and set the elements to the hostnames of the systems on the R1 side of the SRDF link. The order of the hostnames is not important.

d. Leave commented the `R2HOST` array variable and set the elements to the hostnames of the systems on the R2 side of the SRDF link. This will be defined in the recovery cluster later. The order of the hostnames is not important.

e. Uncomment the `RETRY` and `RETRYTIME` variables. The defaults should be used for the first package. The values should be slightly different for other packages. `RETRYTIME` should increase by two seconds for each package. The product of `RETRY * RETRYTIME` should be approximately five minutes. These variables are used to decide how often and how many times to retry the Symmetrix status commands.

For example, if there are three packages that have data on a particular Symmetrix pair (connected by SRDF), then the values for `RETRY` and `RETRYTIME` might be as follows:

<table>
<thead>
<tr>
<th></th>
<th>RETRY</th>
<th>RETRYTIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>pkgA</td>
<td>60</td>
<td>5</td>
</tr>
<tr>
<td>pkgB</td>
<td>43</td>
<td>7</td>
</tr>
<tr>
<td>pkgC</td>
<td>33</td>
<td>9</td>
</tr>
</tbody>
</table>

f. Uncomment the `CLUSTER_TYPE` variable and set it to continental.

8. Edit the remaining control script variables (`VG, LV, FS, IP, SUBNET, SERVICE_NAME, SERVICE_CMD` and `SERVICE_RESTART`) according to the needs of the application as it runs on the primary cluster. See the MC/ServiceGuard manual for more information on these variables.

9. Add customer-defined run and halt commands in the appropriate places according to the needs of the application. See the MC/ServiceGuard manual for more information on these functions.

10. Distribute EMC SRDF package control script files to other nodes in
Setting up a Primary Package on the Primary Cluster

the primary cluster by using ftp or rcp:

```
# rcp -p /etc/cmcluster/<pkg_name>/<pkg_name>.cntl \
other_node:/etc/cmcluster/<pkg_name>/<pkg_name>.cntl
```

When using ftp, be sure to make the file executable on any destination systems.

11. Verify that each host in both clusters in the continental cluster has the following files in the directory /etc/cmcluster/<pkg_name>:
   - <pkg_name>.cntl (EMC SRDF package control script)
   - <pkg_name>.conf (MC/ServiceGuard package ASCII config file)
   - <pkg_name>.sh (Package monitor shell script, if applicable)

12. Split the SRDF logical links for the disks associated with the application package. See the script Samples-CC/pre.cmquery for an example of how to automate this task. The script must be customized with the Symmetrix device group names.

13. Apply the MC/ServiceGuard configuration using the cmapplyconf command or SAM.

14. Using standard MC/ServiceGuard commands (cmruncl, cmhaltcl, cmrunpkg, cmhaltpkg), test the primary cluster for cluster and package startup and package failover.

15. Restore the SRDF logical links for the disks associated with the application package. See the script Samples-CC/post.cmapply (after recovery cluster is completed in next section) for an example of how to automate this task. The script must be customized with the Symmetrix device group names.

The primary cluster is now ready for Continental Cluster operation.
Setting up a Recovery Package on the Recovery Cluster

The installation of EMC SRDF, MC/ServiceGuard, and ContinentalClusters software is exactly the same as in the previous section.

The procedures below will install and configure a recovery package on the recovery cluster. Consult the MC/ServiceGuard documentation for instructions on setting up an MC/ServiceGuard cluster (i.e., lan, VG, LV, etc).

1. Split the EMC/SRDF logical links for the disks associated with the application package. See the script Samples-CC/pre.cmquery for an example of how to automate this task. The script must be edited to refer to the SRDF groups configured and customized with the Symmetrix device group names.

2. Use the following command to generate a cluster ASCII file:
   
   ```
   # cmquerycl -n node1 -n node2 -C CClusterNY.ascii
   ```

   Edit the file **CClusterNY.ascii**. Be sure to select a primary cluster lock disk that is not a lock disk on the recovery cluster. Edits include spreading `HEARTBEAT_IP` on all user LANs, and setting `MAX_PACKAGES`.

3. Check the configuration with the following command:

   ```
   # cmcheckconf -C CClusterNY.ascii
   ```

4. Create the cluster binary with the following command:

   ```
   # cmapplyconf -C CClusterNY.ascii
   ```

5. Test the cluster using the following commands:

   ```
   # cmrunc1 -v
   # cmviewcl -v
   ```

   Does the cluster come up? If so, then stop the cluster:

   ```
   # cmhaltcl -f
   ```

6. Copy the package files from the primary cluster to a bkpkgXXX directory, and rename it to `<backup_pkg_name>.cntl`. Edit the
Physical Data Replication for Continental Clusters Using EMC SRDF

Setting up a Recovery Package on the Recovery Cluster

recovery package control file from the primary cluster for the secondary cluster. Change the subnet, relocatable IP, and nodes.

---

**NOTE**

Remember not to create a MC/ServiceGuard application package control script template with the `cmmakepkg -s` command or by using SAM. The package control script template included with this product should be used instead. (Use the rcp’d files from primary cluster.)

Be sure to set `AUTO_RUN` to `NO` in the package ASCII file.

7. Edit the recovery package control script `<bk_pkg_name>.cntl` as follows:

   a. Add the path for SymCLI software binaries.

   b. Make sure that all `AUTO*` variables are uncommented.

   c. Uncomment the `PKGDIR` variable and set it to the full path name of the directory where the control script has been placed. This directory must be unique for each package and is used for status data files. For example, set `BKPKGDIR` to `/etc/cmcluster/<backup_package_name>`.

   d. Uncomment the `PKGR*` variables.

   e. Leave commented the `R1HOST` array variable.

   f. Uncomment the `R2HOST` array variable and set the elements to the hostnames of the systems on the R2 side of the SRDF link.

   g. Uncomment the `RETRY` and `RETRYTIME` variables. - same

   h. Make sure the `CLUSTER_TYPE` variable is set to `continental`.

   i. Edit the remaining application package control script variables (`VG, LV, FS, IP, SUBNET, SERVICE_NAME, SERVICE_CMD` and `SERVICE_RESTART`) according to the needs of the application. See the MC/ServiceGuard manual for more information on these variables. Change the Subnet IP from ftp copy.

---

**NOTE**

The mount point i.e., `/pkgCCA` and VG are the same in both clusters.

8. Verify that each host in both clusters in the continental cluster has
Physical Data Replication for Continental Clusters Using EMC SRDF

Setting up a Recovery Package on the Recovery Cluster

the following files in the directory /etc/cmcluster/<pkg_name>:

- <pkg_name>.cntl (continental cluster package control script)
- <pkg_name>.conf (MC/ServiceGuard package ASCII config file)
- <pkg_name>.sh (Package monitor shell script, if applicable)

9. Split the SRDF logical links for the disks associated with the application package. See the script Samples-CC/pre.cmquery for an example of how to automate this task. The script must be customized with the Symmetrix device group names.

10. Apply the MC/ServiceGuard configuration using the cmapplyconf command or SAM for the recovery cluster.

11. Test the cluster and packages as follows:

    # cmrunc1
    # cmmodpkg -e bkpkgCCA
    # cmviewcl -v

    Note that cmmodpkg is used to manually start application packages. Do all application packages start? If so, then issue the command:

    # cmhaltcl -f

---

**NOTE**

Application packages cannot run on R1 and R2 at the same time. Any running package on the primary cluster that will have a counterpart on the recovery cluster must be halted to prevent data corruption.

---

12. Restore the SRDF logical links for the disks associated with the application package. See the script Samples-CC/post.cmapply for an example of how to automate this task. The script must be customized with the Symmetrix device group names.

The recovery cluster is now ready for continental cluster operation.
Setting up the Continental Cluster Configuration

The procedures below will configure ContinentalClusters and the monitoring packages on the two clusters. For complete details on creating and editing the configuration file, refer to Chapter 5, “Building a Continental Cluster.”

1. Split the SRDF logical links for the disks associated with the application package. See the script Samples-CC/pre.cmquery for an example of how to automate this task. The script must be customized with the Symmetrix device group names.

2. Generate the ContinentalClusters configuration using the following command:

```bash
# cmqueryconcl -C cmconcl.config
```

3. Edit the configuration file cmcond.config with the names of the two clusters, the nodes in each cluster, the recovery groups and the monitoring definitions. The recovery groups define the primary and recovery packages. Note that when data replication is done using EMC SRDF, there are no data sender and receiver packages.

Define the monitoring parameters, the notification mechanism (ITO, email, console, SNMP, syslog or tcp) and notification type (alert or alarm) based on the cluster status (unknown, down, up or error). Descriptions for these can be found in the configuration file generated in the previous step.

4. Edit the continental cluster security file /etc/opt/cmom/cmomhosts to allow or deny hosts read access by the monitor software.

5. On all nodes in both clusters copy the monitor package files from /opt/cmcond/scripts to /etc/cmdcluster/ccmonpkg. Edit the monitor package configuration as needed in the file /etc/cmdcluster/ccmonpkg/ccmonpkg.config. Set the AUTO_RUN flag to YES. This is in contrast to the flag setting for the application packages. We want the monitor package to start automatically when the cluster is formed.

6. Apply the monitor package to both cluster configurations using the following command:
Setting up the Continental Cluster Configuration

7. Restore the logical SRDF links for the package. See the script Samples-CC/post.cmapply for an example of how to automate this task. The script must be customized with the appropriate Symmetrix device group names. Example:

```
# Samples-CC/post.cmapply
```

8. Start the monitor package on both clusters.

**NOTE**

The monitor package for a cluster checks the status of the other cluster and issues alerts and alarms, as defined in the ContinentalClusters configuration file, based on the other cluster’s status.

9. Check /var/adm/syslog/syslog.log for messages and check for running processes cmclrmond and

```
/opt/cmconcl/jre/bin/PA_RISC/native_threads/jre
```

Also check the ccmonpkg package log file.

10. Generate the cluster configuration file using cmapplyconcl. Files are placed in /var/adm/cmconcl/instances. There is no change to /etc/cmcluster/cmclconfig nor is there an equivalent file for ContinentalClusters. Example:

```
# cmapplyconcl -C cmconcl.config
```

**NOTE**

You must create/edit /.rhosts on all the systems in the continental cluster to allow write access to the system running the cmapplyconcl. After the ContinentalClusters configuration file is successfully generated the file or system entries can be deleted.

11. Start the primary packages on the primary cluster using cmrunpkg.

Test local failover within the primary cluster.

12. View the status of the ContinentalCluster primary and recovery clusters, including configured event data:

```
# cmviewconcl -v
```

The continental cluster is now ready for testing. See “Testing the Continental Cluster” on page 182.
Switching to the Recovery Cluster in Case of Disaster

It is vital the administrator verify that recovery is needed after receiving a cluster alert or alarm. Network failures may produce false alarms.

After validating a failure, start the recovery process using the cmrecovercl \([-f]\) command. Note the following:

- During an alert, the cmrecovercl will not start the recovery packages unless the -f option is used.
- During an alarm, the cmrecovercl will start the recovery packages without the -f option.
- When there is neither an alert nor an alarm condition, cmrecovercl cannot start the recovery packages on the recovery cluster. This condition applies not only when no alert or alarm was issued, but also applies to the situation where there was an alert or alarm, but the primary cluster recovered and its current status is Up.

Verify SRDF links are up:

`# symrdf list`
Failback Scenarios

There is no failback counterpart to the “pushbutton” failover from the primary cluster to the recovery cluster. Failback is very dependent on the original nature of the failover, the state of primary and secondary EMC SRDF volumes (R1 and R2) and the condition of the primary cluster. In Chapter 5, “Building a Continental Cluster,” there is a discussion of failback mechanisms and methodologies in the section “Restoring Disaster Tolerance” on page 190.

The goal of HP ContinentalClusters is to maximize system and application availability. However, even systems configured with ContinentalClusters can experience hardware failures at the primary site or the recovery site, as well as the hardware or networking failures connecting the two sites. The following discussion addresses some of those failures and suggests recovery approaches applicable to environments using data replication provided by EMC Disk Arrays and Symmetrix Remote Data Facility SRDF.

Careful reading of the `srdfpkg.cntl` file will help in general understanding of the package startup. Elaborate SRDF pairstate status checking takes place.

Scenario 1

The primary site has lost power, including backup power (UPS), to both the systems and disk arrays that make up the MC/ServiceGuard Cluster at the primary site. There is no loss of data on either the EMC or the operating systems of the systems at the primary site.

After reception of the ContinentalClusters alerts and alarm, the administrators at the recovery site follow the prescribed processes and recovery procedures to start the protected applications on the recovery cluster. The ContinentalClusters control file will evaluate the status of the R1 primary and R2 paired group volumes.
Physical Data Replication for ContinentalClusters Using EMC SRDF

Failback Scenarios

The command `symrdf list` will display the paired status of the links.

Example--  Dev 001 has failed over  Device Group pkgCCB_r1

Symmetrix ID: 000183500021
Local Device View

<table>
<thead>
<tr>
<th>STATUS</th>
<th>MODES</th>
<th>RDF STATES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sym</td>
<td>RDF</td>
<td>R1 Inv</td>
</tr>
<tr>
<td>Dev</td>
<td>RDev</td>
<td>Tracks</td>
</tr>
<tr>
<td>---</td>
<td>----</td>
<td>---</td>
</tr>
<tr>
<td>000</td>
<td>000</td>
<td>0 WD RW</td>
</tr>
<tr>
<td>001</td>
<td>001</td>
<td>12 RW WD</td>
</tr>
<tr>
<td>002</td>
<td>002</td>
<td>0 WD RW</td>
</tr>
<tr>
<td>003</td>
<td>003</td>
<td>0 WD RW</td>
</tr>
<tr>
<td>004</td>
<td>004</td>
<td>0 WD RW</td>
</tr>
<tr>
<td>005</td>
<td>005</td>
<td>0 WD RW</td>
</tr>
<tr>
<td>006</td>
<td>006</td>
<td>0 WD RW</td>
</tr>
<tr>
<td>007</td>
<td>007</td>
<td>0 WD RW</td>
</tr>
</tbody>
</table>

After power is restored to the primary site, the EMC device groups may have either a status of Failed Over. The procedure to move the application packages back to the primary site are different depending on the status of the device groups at the primary site.

The following procedure applies to the situation where the primary site device groups have a status of “Failed Over”:

1. Halt the ContinentalClusters recovery packages at the recovery site using the following command:
   
   `# cmhaltpkg <pkg_name>`
   
   This will halt any applications, remove any floating IP addresses, unmount file systems and deactivate volume groups as programmed into the package control files. The status of the device groups will remain “Synchronized” at the recovery site and “Failed Over” at the primary site.

2. Halt the cluster, which also halts the monitor package `ccmonpkg`.

3. Start the cluster at the primary site. Assuming they have been properly configured the ContinentalClusters primary packages should not start. The monitor package should start automatically.

4. Manually start the ContinentalClusters primary packages at the primary site using
   
   `# cmrunpkg <pkg_name>`
   
   or
Physical Data Replication for ContinentalClusters Using EMC SRDF

Failback Scenarios

# cmmodpkg -e <pkg_name>

The control script is programmed to handle this case. The control script will issue an SRDF failback command to move the device group back to the R1 side and to resynchronize the R1 from the R2 side. Until the resynchronization is complete, the SRDF "read-through" feature will ensure that any reads on the R1 will be current, by reading data through the ESCON link from the R2 side.

NOTE
If the system administrator does not want synchronization performed from the remote (recovery) site, the device groups should be split and recreated manually.

5. Ensure that the monitor packages at the primary and recovery sites are running.

6. Issue the following command:

   # symrdf list

to verify device group is synchronized.

7. If the package does not come up, and the device group status is “failed over,” then manually bring it back using the command: symrdf -g <groupname_r1> failback. Example:

   # symrdf -g pkgCCB_r1 failback

   Execute an RDF 'Failback' operation for device group 'pkgCCB_r1' (y/[n]) ? y

   An RDF ‘Failback’ operation execution is in progress for device group ‘pkgCCB_r1’.

   Write Disable device(s) on RA at target (R2) ..............Done.
   Suspend RDF link(s)......................................Done.
   Merge device track tables between source and target....Started.
   Device: 001 ............................................. Merged.
   Merge device track tables between source and target....Done.
   Resume RDF link(s)......................................Done.
   Read/Write Enable device(s) on SA at source (R1)...Done.

   The RDF ‘Failback’ operation successfully executed for device group ‘pkgCCB_r1’.

8. During the resync it goes from status failed over > invalid > SyncInProg. Example:
Physical Data Replication for ContinentalClusters Using EMC SRDF

Failback Scenarios

ftsys1a# symrdf list

Symmetrix ID: 000183500021

Local Device View

<table>
<thead>
<tr>
<th>STATUS</th>
<th>MODES</th>
<th>RDF</th>
<th>STATES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sym</td>
<td>RDF</td>
<td>------</td>
<td>R1 Inv</td>
</tr>
<tr>
<td>Dev</td>
<td>RDev</td>
<td>Typ:G</td>
<td>SA</td>
</tr>
<tr>
<td>---</td>
<td>----</td>
<td>-----</td>
<td>-------</td>
</tr>
<tr>
<td>000</td>
<td>000</td>
<td>R2:2</td>
<td>RW</td>
</tr>
<tr>
<td>001</td>
<td>001</td>
<td>R2:2</td>
<td>RW</td>
</tr>
</tbody>
</table>

ftsys1a# symrdf list

Symmetrix ID: 000183500021

Local Device View

<table>
<thead>
<tr>
<th>STATUS</th>
<th>MODES</th>
<th>RDF</th>
<th>STATES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sym</td>
<td>RDF</td>
<td>------</td>
<td>R1 Inv</td>
</tr>
<tr>
<td>Dev</td>
<td>RDev</td>
<td>Typ:G</td>
<td>SA</td>
</tr>
<tr>
<td>---</td>
<td>----</td>
<td>-----</td>
<td>-------</td>
</tr>
<tr>
<td>000</td>
<td>000</td>
<td>R2:2</td>
<td>RW</td>
</tr>
<tr>
<td>001</td>
<td>001</td>
<td>R2:2</td>
<td>RW</td>
</tr>
</tbody>
</table>

9. Halt the recovery cluster and restart it:

   # cmhaltcl -f (if not already down)
   # cmruncl

10. Verify the data for data consistency and currency.
Scenario 2

The primary site EMC experienced a catastrophic hardware failure and all data was lost on the array.

After reception of the ContinentalClusters alerts and alarm, the administrators at the recovery site follow prescribed processes and recovery procedures to start the protected applications on the recovery cluster. The ContinentalClusters control file will evaluate the status of the EMC paired volumes. Since the systems at the primary site are accessible, but the EMC is not, the control file will evaluate the paired volumes with a local status of failed over. The control file script is programmed to handle this condition and will enable the volume groups, mount the logical volumes, assign floating IP addresses and start any processes as coded into the script.

After the primary site EMC is repaired and configured appropriately use the following procedure to move the application package back to the primary site.

1. Manually create the EMC device groups and gatekeeper configurations device groups. Rerun the scripts `mk3symgrps*` and `mk4gatekpr*` which do the following:
   
   # date >ftsys1.group.list
   # symdg create -type RDF1 pkgCCA_r1
   # symld -g pkgCCA_r1 add pd /dev/rdsk/c7t0d0
   # symgate define pd /dev/rdsk/c7t15d0
   # symgate define pd /dev/rdsk/c7t15d1
   # symgate -g pkgCCA_r1 associate pd /dev/rdsk/c7t15d0

2. Halt the ContinentalClusters recovery packages at the recovery site using

   # cmhaltpkg <pkg_name>

   This will halt any applications, remove any floating IP addresses, unmount file systems and deactivate volume groups as programmed into the package control files. The status of the paired volumes will be SPLIT at both the recovery and primary sites.

3. Halt the Cluster, which also halts the monitor package `ccmonpkg`. 
4. Start the cluster at the primary site. Assuming they have been properly configured the ContinentalClusters primary packages should not start. The monitor package should start automatically. Since the paired volumes have a status of SPLIT at both the primary and recovery sites, the EMC views the two halves as unmirrored.

5. Issue the following command:

```bash
# symrdf -g pkgCCB_r1 failback
```

Since the most current data will be at the remote or recovery site, this command to synchronize from the remote site. Wait for the synchronization process to complete before progressing to the next step. Failure to wait for the synchronization to complete will result in the package failing to start in the next step.

6. Manually start the ContinentalClusters primary packages at the primary site using

```bash
# cmrunpkg <PKG_NAME>
```

The control script is programmed to handle this case. The control script recognizes that the paired volume is synchronized and will proceed with the programmed package startup.

7. Perform

```bash
# symrdf list
```

to verify the device group is synchronized.

8. Ensure that the monitor packages at the primary and recovery sites are running.
Maintaining the EMC SRDF Data Replication Environment

Normal Startup

The following is the normal ContinentalClusters startup procedure. On the primary cluster:

1. Start the primary cluster:
   
   # cmrunc1 -v

   The primary cluster comes up with ccmonpkg up. The application packages are down, and ccmonpkg is up.

2. Manually start application packages on the primary cluster:
   
   # cmmod pkg -e <Application_pkgname>

3. Confirm primary cluster status:
   
   # cmviewcl -v

   and

   # cmviewconcl -v

4. Verify SRDF Links:

   # symrdf list

On the recovery cluster, do the following:

1. Start the recovery cluster with the following commands:

   # cmrunc1 -v

   The recovery cluster comes up with ccmonpkg up. The application packages (bkpkgX) stay down, and ccmonpkg is up.

2. Do not manually start application packages on the recovery cluster; this will cause data corruption.

3. Confirm recovery cluster status:

   # cmviewcl -v

   and
Maintaining the EMC SRDF Data Replication Environment

```
# cmviewconcl -v
```

Normal Maintenance

There might be situations where a package has to be taken down for maintenance purposes without having the package move to another node.

The following procedure is recommended for normal maintenance of the ContinentalCluster with EMC SRDF data replication:

1. Shut down the package with the appropriate command. Example:
   ```
   # cmhaltpkg <pkgname>
   ```

2. Distribute the ContinentalClusters configuration changes. Example:
   ```
   # cmapplyconf -P <pkgconfig> (Primary cluster)
   # cmapplyconf -P <bkpkgconfig> (Recovery cluster)
   ```

3. Start up the package with the appropriate MC/ServiceGuard command. Example:
   ```
   # cmmodpkg -e <pkgname> (Primary cluster)
   ```

   **NOTE**
   Never enable package switching on both the primary package and the recovery package.

4. Halt the monitor package:
   ```
   # cmhaltpkg ccmonpkg
   ```

5. To apply the new continental cluster configuration:
   ```
   # cmapplyconcl -C <configfile>
   ```

6. Restart the monitor package:
   ```
   # cmrunpkg ccmonpkg
   ```

Some Further Points

Following are listed some EMC Symmetrix-specific requirements:

- R1 devices must be locally protected (RAID 1 or RAID S); R2 devices must be locally protected (RAID 1, RAID S or BCV)
Physical Data Replication for ContinentalClusters Using EMC SRDF

Maintaining the EMC SRDF Data Replication Environment

• Only synchronous mode is supported; adaptive copy must be disabled.
  
  Domino Mode is recommended to ensure that there is no possibility of inconsistent data at the R2 side in case of SRDF link failure. Since only dedicated ESCON links are supported, the probability of intermittent link failure is extremely low. Therefore, the probability of inconsistent data at the R2 side is extremely low.
  
  Domino Mode will prevent the data from becoming inconsistent on the R2 side if all ESCON links between one R1/R2 Symmetrix pair is lost, or one of the R2 side Symmetrix frames fails.
  
  If Domino Mode is not enabled and the SRDF link fails and the application continues to modify data and the link is restored and resynchronization from R1 to R2 starts (but doesn't finish) and the R1 side fails, then the data at the R2 side are inconsistent and therefore unusable.

  Although the risk of this occurrence is extremely low, if your business cannot afford even this quite small risk, then you must enable Domino Mode to ensure that the data at the R2 side are always consistent.

  The disadvantage of enabling Domino Mode is that when the SRDF link fails, all I/Os will be refused (to those devices) until the SRDF link is restored, or manual intervention is undertaken to disable Domino Mode. Applications may fail or may continuously retry the I/Os (depending on the application) if Domino Mode is enabled and the SRDF link fails.

• SRDF firmware has been configured and hardware has been installed on both Symmetrix units
  
• R1 and R2 devices must be correctly defined and assigned to the appropriate host systems in the internal configuration that is downloaded by EMC
  
• While the cluster is running, all Symmetrix devices that belong to the same MC/ServiceGuard package, and defined in a single SRDF group must be in the same state at the same time. Manual changes of these states can cause the package to halt due to unexpected conditions. In general, it is recommended that no manual change of states be performed while the package and the cluster are running.

• A single Symmetrix device group must be defined for each package on each host that is connected to the Symmetrix. The disk special device file names for all Volume Groups that belong to the package
Physical Data Replication for ContinentalClusters Using EMC SRDF

Maintaining the EMC SRDF Data Replication Environment

must be defined in one Symmetrix device group on the R1 side and one Symmetrix device group on the R2 side.

The R1 group names must be the same on each host on the R1 side and the R2 group names must be the same on each host on the R2 side. These group names are placed in the shell script array variables PKGR1SYMGRP and PKGR2SYMGRP.

Although the name of the device group must be the same on each node, the special device file names specified may be different on each node.

Symmetrix Logical Device names MUST be default names of the form “DEVnnn” (e.g., DEV001). Do not use the option for creating your own device names.

See the SymCLI manual, and the sample convenience scripts in the Samples-CC directory included with this toolkit.

• To minimize contention, each device group used in the package should be assigned two unique gatekeeper devices on the Symmetrix for each host where the package will run. These gatekeeper devices must be associated with the Symmetrix device groups for that package. The gatekeeper devices are typically a 2 MB logical device on the Symmetrix.

For example, if a package is configured to failover across four nodes in the cluster, there should be eight gatekeeper devices (two for each node) that are assigned to the Symmetrix device group belonging to this package.

It is required that there be a pool of four additional gatekeeper devices that are NOT associated with any device group. These gatekeepers would be available for other, non-cluster uses, e.g., the Symmetrix Manager GUI and other SymCLI or SymAPI requests.

After data configuration, each physical device in the Symmetrix has enough space remaining on it for gatekeeper purposes.

• This toolkit does not support the HP OmniBack Integration with Symmetrix. The OmniBack Integration with Symmetrix may create certain states that will cause this package to halt if a failover occurs while the backup is in progress.

• No checking of the status of the SA/FA ports is done. It is assumed that at least one PVLink is functional. Otherwise, the VG activation will fail.
Physical Data Replication for Continental Clusters Using EMC SRDF

Maintaining the EMC SRDF Data Replication Environment

• This toolkit may increase package startup time by 5 minutes or more. Packages with many disk devices will take longer to start up than those with fewer devices due to the time needed to get device status from the Symmetrix. Clusters with multiple packages that use devices on the Symmetrix will all cause package startup time to increase when more than one package is starting at the same time.

• The value of `RUN_SCRIPT_TIMEOUT` in the package ASCII file should be set to `NO_TIMEOUT` or to a large enough value to take into consideration the extra startup time due to getting status from the Symmetrix. See the previous paragraph for more information on the extra startup time.
8 Cascading Failover in a Continental Cluster

This chapter shows how to create a configuration that allows cascading failover between two sites that are connected via Continental Clusters. Cascading failover is the ability of an application to fail from a primary to a secondary location, and then to fail to a recovery location on a different site. The primary location contains a metropolitan cluster built with MetroCluster EMC SRDF, and the recovery location has a standard MC/ServiceGuard cluster. The solution will support up to 16 nodes in the primary cluster and 16 nodes in the recovery cluster.

This chapter has the following sections:

- Overview
- Data Storage Setup
- Primary Cluster Package Setup
- Recovery Cluster Package Setup
- Continental Cluster Configuration
- Data Replication Procedures

It is assumed that readers are already familiar with MC/ServiceGuard configuration tasks, continental cluster installation and configuration procedures, EMC Symmetrix, BCV, SRDF, Symmetrix multi-hop concepts, and Symmetrix Command Line Interface (SymCLI) configuration and use.

NOTE This chapter is only for users of Continental Clusters with the EMC Symmetrix disk array.
Overview

The basic design of the cascading failover solution is shown in Figure 8-1. The primary cluster, shown on the left, is configured as a metropolitan cluster with three data centers physically located on three different sites—two main sites (Site 1 and Site 2) and an arbitrator site (Site 3). A fourth data center (Site 4) is used for the recovery cluster, which is a standard MC/ServiceGuard configuration.

Figure 8-1  Continental Cluster with Cascading Failover

In the primary cluster, there are two Symmetrix frames, either of which can have R1 volumes for a particular application. Throughout this document, the term primary Symmetrix refers to the Symmetrix that holds R1 volumes for a particular application, and the data center where this Symmetrix is located is called the primary site. The term secondary Symmetrix refers to the Symmetrix that holds the R2
volumes for a particular application, and the data center where the secondary Symmetrix for that application is located is known as the secondary site. Thus, primary and secondary are roles that can be played by either Symmetrix in the primary cluster. In Figure 8-1, Symm1 holds the R1 volumes for application X data, and Site 1 is known as the primary site for application X. Symm2 holds R1 volumes for application Y data, and Site 2 is the primary site for application Y.

The recovery Symmetrix holds a remote replicated copy of the data in the recovery cluster. The data center that houses the recovery Symmetrix is called the recovery site. In Figure 8-1, Symm3 is the recovery Symmetrix for both application X data and application Y data, and Site 4 is the recovery site for both application X and application Y.

Symmetrix Configuration

There are three EMC Symmetrix disk arrays. Each has a minimum of two 2-port RDF Director boards. The basic configuration requires the following in both Symmetrixes in the primary cluster:

- R1 volumes in the primary Symmetrix to hold the primary data for an application
- R2 volumes in the secondary Symmetrix to hold a current copy of the data for an application
- Business Continuity volumes (BCV’s) in the secondary Symmetrix to be used as the intermediary volumes to replicate the data to the recovery Symmetrix
- R2 volumes in the recovery Symmetrix to hold a point in time copy of the data
- BCV’s in the recovery Symmetrix to protect the data from corruption due to rolling disaster
- Two ESCON connections over WAN using ESCON converters between the secondary Symmetrix and the recovery Symmetrix
- Two direct ESCON connections (ESCON extenders are needed if the distance is over 3 kilometers) between the primary Symmetrix and the secondary Symmetrix

The Symmetrix configuration is shown in Figure 8-2.
Cascading Failover in a Continental Cluster

Overview

Figure 8-2  Symmetrix Configuration for Cascading Failover

Using Template Files

Cascading failover configurations are built using the EMC SRDF control script template, /opt/cmcluster/toolkit/SGSGRDF/srdfpkg.cntl. This file should be used on all ContinentalClusters nodes and clusters that:

- will run a package in the primary MC/ServiceGuard cluster and
- will run a recovery package in the recovery MC/ServiceGuard cluster and
- whose data are on an EMC Symmetrix ICDA connected to the primary cluster and replicated with SRDF to a second EMC Symmetrix ICDA also connected to the primary cluster, and then replicated with SRDF to a third EMC Symmetrix ICDA connected to the recovery cluster

In addition to the basic template, there is a set of sample scripts that illustrate how to carry out most of the procedures described in this chapter. These scripts will be found in the /opt/cmcluster/toolkit/SGSRDF/cascade/Samples directory.

To use these files, follow the procedures detailed in the next sections.
Data Storage Setup

These procedures will prepare the MC/ServiceGuard clusters for use in a ContinentalClusters Cascading Failover configuration.

Setting Up Symmetrix Device Groups

Work with your EMC support engineer to configure the devices in three Symmetrix disk arrays. Figure 8-3 shows an example of the Symmetrix devices setup for an application that runs on clear.

Use the following steps:

1. By default, the SRDF link is configured with automatic link recovery after all the failure of all links. Work with EMC support engineers to disable this feature by setting the flag "Prevent Automatic Links Recovery after all Links Failure" in the Symmetrix bin file to "Yes." This will keep the links from automatically trying to establish a new connection upon failure of all links. This is required so the user
Cascading Failover in a Continental Cluster

Data Storage Setup

would have time to split the BCV/R1 devices in the secondary Symmetrix from the mirror group before re-establishing the SRDF volume pairs between the primary Symmetrix and the secondary Symmetrix upon link recovery.

2. Ensure that the Symmetrix ICDAs are correctly cabled to each host system in the cluster that will run packages whose data reside on the Symmetrix.

3. Install the SymCLI software on each host system that has data residing on the Symmetrix. Using Figure 8-3, the SymCLI software is installed on hosts clear, erie, newyork, seattle, atlanta and sanfran.

4. Create or rebuild the SymCLI database on each system with the most current information. Use the command:
   
   `# symcfg discover`

5. Get a list of data for the Symmetrix devices visible to the host. Use the command:

   `# syminq -sym`

   to get a list that includes the following data for each Symmetrix logical device:

   - HP-UX special device filename
   - Device type (R1, R2, BCV or blank)
   - Symmetrix serial number

   Gatekeeper devices are usually of size 2880. The Symmetrix device serial number is useful in matching the devices to the actual devices in the Symmetrix configuration (downloaded by EMC):

   - The first two digits equal the last two digits of the serial number of the Symmetrix frame. The first two digits equal the last two digits of the serial number of the Symmetrix frame.
   - The next three hexadecimal digits equal the Symmetrix device number that is seen in the output of the following command:

     `# symdev list`

     - The next three digits equal the Symmetrix host adapter (SA or FA) and port numbers; this is useful to see multiple host links to the same Symmetrix device (PVLinks).

6. The BCV devices are configured to be visible on a different SA/FA
than R1 or R2 devices. These devices will not show on the output of the `syminq` command unless there is a host connection to this SA/FA and the BCV’s are split from the mirror group. To get a list of these devices, use the command:

```
# symdev list
```

7. Get a list of the disk devices known to the host system. Use the following command, as appropriate:

```
# ioscan -kfnC disk
```

or

```
# ioscan -fnC disk
```

Match the device file names in this listing with the device file names in the output from the `syminq` command to see which devices are seen from this host to make sure that this host can see all necessary devices.

8. There are three Symmetrix device groups needed for each application package:

- One device group contains the R1 devices in the primary Symmetrix plus its remote BCV devices in the secondary Symmetrix. This device group has to be defined in all nodes that connect to the primary Symmetrix. Based on the example shown in Figure 8-3, Figure 8-4 shows the devices that need to be included in the primary Symmetrix device group.

![Figure 8-4 Primary Device Group](image)
Cascading Failover in a Continental Cluster

Data Storage Setup

- A second device group contains the R2 devices in the secondary Symmetrix and its local BCV/R1 devices in the same Symmetrix. This device group has to be defined in all nodes that connect to the secondary Symmetrix. Figure 8-5 shows the devices that need to be included in the secondary Symmetrix device group.

Figure 8-5  Secondary Device Group

- A third device group contains the R2 devices in the recovery Symmetrix and its local BCV devices in the same Symmetrix. This device group has to be defined in all nodes that connect to the recovery Symmetrix. Figure 8-6 shows the devices that need to be included in the recovery Symmetrix device group.

Figure 8-6  Recovery Device Group
Use the `symdg`, `symld`, `symbcv`, and `symgate` commands on all nodes in both clusters to create the Symmetrix device groups. All devices belonging to Volume Groups that are owned by an application package must be defined in a single Symmetrix device group.

A unique Symmetrix gatekeeper device should be created for each application package. The gatekeeper devices must be unique per MC/ServiceGuard package to prevent contention in the Symmetrix when commands are issued. Gatekeeper devices are unique to a Symmetrix unit. They are not replicated across the SRDF link. There should be at least two gatekeeper devices accessible from each host system. Each gatekeeper device is configured on different physical links. The SymCLI will switch to an alternate gatekeeper device if the path to the primary device fails.

See the scripts in the `/opt/cmcluster/toolkit/SGSRDF/cascade/Samples` directory for examples of how these commands are used to build the SymCLI database and setup the Symmetrix device groups. The sample scripts are used as follows:

- `mkprigrp`—create device group on nodes that connect to the primary Symmetrix. Run this script only on each node that connects to the primary Symmetrix.
- `mksecgrp`—create device group on nodes that connect to the secondary Symmetrix. Run this script only on each node that connects to the secondary Symmetrix.
- `mkrecgrp`—create device group on nodes that connect to the recovery Symmetrix. Run this script only on each node that connects to the recovery Symmetrix.

After the device groups are created, the SymCLI database can be viewed with the commands:

- List all device groups:
  
  `symdg list`

- List the content of a device group:
  
  `symld -g <symdevgrpname> list`

- List the gatekeeper devices:
  
  `symgate list`

- List all devices and their states:
Cascading Failover in a Continental Cluster

Data Storage Setup

#symrdf list

9. Create a text file that contains a list of the standard device and BCV device pair in the secondary Symmetrix. This file is required when doing operations on the BCV devices in the secondary Symmetrix from a node that connects to the recovery Symmetrix.

This file has two columns. The first column contains the Symmetrix device number for the standard device. The second column contains the Symmetrix device number for the BCV device. Refer to the sample file midhop in the /opt/cmcluster/toolkit/SGSRDF/cascade/Samples directory.

10. Create a text file that contains a list of the standard device and BCV device pair in the recovery Symmetrix. This file is required when doing operations on the BCV devices in the recovery Symmetrix from a node that connects to the primary Symmetrix or the secondary Symmetrix. Refer to the sample file lasthop in the /opt/cmcluster/toolkit/SGSRDF/cascade/Samples directory.

Setting up Volume Groups

Use the following procedure to set up volume groups for the volumes on the Symmetrix frames.

1. Before creating the volume groups, make sure that the SRDF link is established between the R1 devices in primary Symmetrix and the R2 devices in the secondary Symmetrix, and the BCV/R1 devices in the secondary Symmetrix are established as mirrors of the local standard devices. On a node that connects to the primary Symmetrix, use the commands:

#symrdf -g <prismdevgrpname> est -v
#symmir -g <prismdevgrpname> -full est -rdf

2. Define the appropriate volume groups on each host system that might run the application package. Use the commands:

#mkdir /dev/vgxx
#mknod /dev/vgxx/group c 64 0xnn0000

where the name /dev/vgxx and the number nn are unique within the entire continental cluster.

3. Create the volume group only on one cluster node that connects to the
4. The LVM information needs to be replicated to the R2 devices in the recovery Symmetrix. See the script `copytorec` in the `/opt/cmcluster/toolkit/SGSRDF/cascade/Samples` directory for an example of how the replication is done. From a node that connects to the primary Symmetrix, run the script or do the following:

   a. Split the BCV/R1 devices in the secondary Symmetrix from their mirror groups:

      ```bash
      # symmir -g <prisymdevgrpname> split -rdf
      ```

   b. Fully establish the SRDF link between the BCV/R1 devices in secondary Symmetrix and the R2 devices in the recovery Symmetrix:

      ```bash
      # symrdf -g <prisymdevgrpname> -full est -rbcv
      ```

   c. Use the following command to check the data synchronization from the BCV/R1 devices to the R2 devices. If the “RDF Pair STATE” column shows the state “Synchronized” for all the devices, the copying completed.

      ```bash
      # symrdf -g <prisymdevgrpname> query -rbc
      ```

   d. Once the copy completes, split the SRDF link between the secondary Symmetrix and the recovery Symmetrix:

      ```bash
      # symrdf -g <prisymdevgrpname> split -rbcv
      ```

   e. Fully establish the BCV devices in the recovery Symmetrix as mirrors of the standard devices:

      ```bash
      # symmir -f <bcvdev_textfile> -sid <recsymid> -full est
      ```

      The `<bcvdev_textfile>` is a file that contains a list of the standard device and BCV device pair in the recovery Symmetrix.

   f. Re-establish the BCV/R1 devices to the mirror group as a mirror of the standard device. This would make the devices invisible to the systems so the `vgimport` would work properly. Use the command:

      ```bash
      # symmir -f <prisymdevgrpname> est -rdf
      ```

5. Generate a map file by exporting the VGs on the node that was used to create the VGs without removing the special device files. Uses the commands:
Cascading Failover in a Continental Cluster

Data Storage Setup

```
# vgchange -a n <vgname>
# vgexport -s -p -m <mapfilename <vgname>
Make sure to copy the mapfiles to all of the host systems.
```

6. Import the VGs on all of the other systems that might run the
MC/ServiceGuard package and back up the LVM configuration. Do
this on both the primary and recovery cluster nodes. Make sure the
SRDF link between the primary Symmetrix and the secondary
Symmetrix is split before importing the VGs.

Use the following commands:

```
# symrdf -g <prisymdevgrpname> split -v
# vgimport -s -m <mapfilename> <vgname>
# vgchange -a y <vgname>
# vgcfgbackup <vgname>
# vgchange -a n <vgname>
# symrdf -g <prisymdevgrpname> est -v
```

Testing the Volume Groups

Before MC/ServiceGuard and Continental Clusters can be configured,
HP-UX must be satisfied. Use the following procedure to confirm that
the disks have been shared, and that the volume groups can be seen on
the primary and adoptive nodes:

1. Confirm the volume groups have been imported using command:

   ```
   # strings /etc/lvmtab
   ```

2. Split the SRDF links between the primary Symmetrix and the
   secondary Symmetrix (if linked). Issue the following command from a
   node that connects to the primary Symmetrix:

   ```
   # symrdf -g <prisymdevgrpname> split -v
   ```

3. Activate the volume group on the primary node in the primary
   cluster, and mount the logical volumes.

   ```
   # vgchange -a y <vgname>
   # mkdir <tempdir>
   # mount /dev/<vgname>/<lvolname> <tempdir>
   ```
Check the mount points:

```
# bdf
```

Do they appear? If yes, then deactivate the volume groups:

```
# umount <tempdir>
# vgchange -a n <vgname>
```

4. Perform step 3 on all adoptive nodes in the primary cluster and in the recovery cluster.
Cascading Failover in a Continental Cluster

Primary Cluster Package Setup

This section only describes the required changes to the package control script to support the cascading failover configuration. To create a primary package on the primary cluster, the reader should follow the procedure in “Setting up a Primary Package on the Primary Cluster” on page 244.

Cascading failover uses a ContinentalClusters in which the primary cluster is configured as a metropolitan cluster. Therefore, there are a few differences from the normal ContinentalClusters configuration in customizing the package control script `<package_name>.cntl`.

1. The `CLUSTER_TYPE` variable should be set to "metro"
2. The `PKGR1SYMGRP` variable should be set to the Symmetrix device group name that was defined in the primary Symmetrix.
3. The `PKGR2SYMGRP` variable should be set to the Symmetrix device group name that was defined in the secondary Symmetrix.
4. The elements of the `R1HOST` array variable should be set to the hostnames of the systems that connect to the primary Symmetrix.
5. The elements of the `R2HOST` array variable should be set to the hostnames of the systems that connect to the secondary Symmetrix.

Before executing the `cmapplyconf` on the primary cluster, make sure to split the SRDF logical links, between the primary Symmetrix and the secondary Symmetrix, for the disks associated with the application package. Run the following command on a node that connects to the primary Symmetrix:

```
# symrdf -g <prisymdevgrpname> split -v
```
Cascading Failover in a Continental Cluster
Recovery Cluster Package Setup

**Recovery Cluster Package Setup**

This section only describes the required changes to the package control script to support the cascading failover configuration. To create a recovery package on the recovery cluster, the reader should follow the procedure in “Setting up a Recovery Package on the Recovery Cluster” on page 248. There is a slight difference from the normal ContinentalClusters configuration in customizing the recovery package control script `<bk_package_name>.cntl`:

1. Leave commented the `PKGR1SYMGRP` variable.
2. The `PKGR2SYMGRP` variable should be set to the Symmetrix device group name that was defined in the recovery Symmetrix.
3. The elements of the `R2HOST` array variable should be set to the hostnames of the systems that connect to the recovery Symmetrix.
4. The `CLUSTER_TYPE` variable should be set to “continental”

Before running `cmapplyconf` command for the recovery cluster, make sure to split the SRDF logical links, between the secondary Symmetrix and the recovery Symmetrix, for the disks associated with the application package. Run the following command on a node that connects to the primary Symmetrix:

```
# symrdf -g <prisymdevgrpname> split -rbcv -v
```
Continental Cluster Configuration

To set up the Continental Cluster, the reader should follow the procedure in “Setting up the Continental Cluster Configuration” on page 251. Step 1 in this section calls for a split of SRDF logical links; this is the link between the secondary Symmetrix and the recovery Symmetrix.
Data Replication Procedures

This section describes the procedures that should be used to manage the data in a Symmetrix cascading failover configuration.

Data Initialization Procedures

The following procedures are needed if you already have existing data prior to implementing this solution. If you already have a metropolitan cluster (that is, the primary Symmetrix and the secondary Symmetrix are up and running, and the SRDF volume pairs between the two Symmetrix frames are already established) and you are now adding the recovery cluster to the configuration, only procedure 2 is required.

1. Mirroring from the Primary to the Secondary Symmetrix

This procedure is illustrated in Figure 8-7.

Figure 8-7  Mirroring from the Primary to the Secondary Symmetrix

Execute the following commands from a node that connects to the primary Symmetrix:

1. Incrementally establish the BCV/R1 devices in the secondary Symmetrix as mirrors of the standard devices if they are not already established:

   # symmir -g <prisymdevgrpname> est -rdf

2. Incrementally establish the SRDF volume pairs between the primary Symmetrix and the secondary Symmetrix:

   # symrdf -g <prisymdevgrpname> est
Cascading Failover in a Continental Cluster

Data Replication Procedures

3. Start the application to load the data to the R1 devices in the primary Symmetrix.

Refer to the sample script `datainit` in the `/opt/cmcluster/toolkit/SGSRDF/cascade/Samples` directory for examples of how these commands are used. This script is designed to run only on a node that connects to the primary Symmetrix.

2. Mirroring from the Secondary to the Recovery Symmetrix

This procedure is illustrated in Figure 8-8.

Execute the following commands from a node that connects to the primary Symmetrix:

1. Freeze the application I/O to prevent the application from writing data to the R1 devices in the primary Symmetrix (in case of adding the recovery cluster to the existing MetroCluster). The method of freezing the I/O is application dependent.

2. Split the BCV/R1 devices in the secondary Symmetrix from the mirror group

   `#symmir -g <prisymdevgrname> split -rdf`

3. Resume the application I/O to the R1 devices in the primary Symmetrix if needed. The method of resuming the I/O is application dependent.

4. Incrementally establish the SRDF volume pairs between secondary Symmetrix and the recovery Symmetrix. The data is copied from BCV/R1 devices on the secondary Symmetrix to the R2 devices on the recovery Symmetrix.
Chapter 8 283

Cascading Failover in a Continental Cluster

Data Replication Procedures

#symrdf -g <prisymdevgrpname> est -rbcv

5. Use the following command to check the data synchronization from the BCV/R1 devices to the R2 devices. If the “RDF Pair STATE” column shows the state “Synchronized” for all the devices, the copying completed.

#symrdf -g <prisymdevgrpname> query -rbcv

6. Once the copy completes, split the SRDF link between the secondary Symmetrix and the recovery Symmetrix

#symrdf -g <prisymdevgrpname> split -rbcv

7. Incrementally establish the BCV devices in the recovery Symmetrix as mirrors of the standard devices if they are not already established. These BCV devices were fully established when the volume group were created.

#symmir -f <recbcvdev_textfile> -sid <recsymid> est

The <recbcvdev_textfile> is a file that contains a list of the standard device and BCV device pair in the recovery Symmetrix.

8. Re-establish the BCV/R1 devices in the secondary Symmetrix as mirrors of the standard devices:

#symmir -g <prisymdevgrpname> est -rdf

Refer to the sample script prirefreshrec in the /opt/cmcluster/toolkit/SGSRDF/cascade/Samples directory for examples of how these commands are used. This script is designed to be run only on a node that connects to the primary Symmetrix.

Data Refresh Procedures in the Steady State

Once the application starts writing data to the primary Symmetrix devices, the data on the recovery Symmetrix is out of sync with the primary data; the data is not current but consistent. The primary data needs to periodically synchronized to the recovery Symmetrix so its data is not too out of date. As long as the application continues writing new data to the primary Symmetrix, the data on the recovery Symmetrix will always be behind. The level of data currency on the recovery Symmetrix is dictated by the frequency at which it is refreshed. The refresh process is shown in Figure 8-9.
Cascading Failover in a Continental Cluster

Data Replication Procedures

**Figure 8-9 Data Refresh in Steady State**

The following procedure describes the steps necessary to periodically copy the data from the secondary Symmetrix to the recovery Symmetrix while the application is running on the primary site.

1. Freeze the application I/O to prevent the application from writing data to the R1 devices in the primary Symmetrix. The method of freezing the I/O is application dependent.

2. Use the following command to check the state of the SRDF link from the R1 devices in the primary Symmetrix to the R2 devices in the secondary Symmetrix. Make sure that the state is “Synchronized” before executing step 3.

   ```
   # symrdf -g <prisymdevgrpname> query
   ```

3. Split the BCV/R1 devices in the secondary Symmetrix from the mirror group:

   ```
   # symmir -g <prisymdevgrpname> split -rdf
   ```

4. Resume the application I/O to the R1 devices in the primary Symmetrix if needed. The method of resuming the I/O is application dependent.

5. Split the BCV devices in the recovery Symmetrix from the mirror group. This is required to preserve an old copy of the data just in case a failure occurs during data synchronization between the secondary Symmetrix and the recovery Symmetrix that may cause data corruption on the R2 devices in the recovery Symmetrix.

   ```
   # symmir -f <recbcvdev_textfile> -sid <recsymid> split
   ```
The <recbcvdev_textfile> is a file that contains a list of the standard device and BCV device pair in the recovery Symmetrix.

6. Incrementally establish the SRDF volume pairs between the secondary Symmetrix and the recovery Symmetrix. The data is copied from secondary Symmetrix BCV/R1 devices to the recovery Symmetrix R2 devices:

```
#symrdf -g <prisymdevgrpname> est -rbcv
```

7. Use the following command to check the data synchronization from the BCV/R1 devices to the R2 devices. If the "RDF Pair STATE" column shows the state "Synchronized" for all the devices, the copying completed:

```
#symrdf -g <prisymdevgrpname> query -rbcv
```

8. Once the copy completes, split the SRDF link between the secondary Symmetrix and the recovery Symmetrix:

```
#symrdf -g <prisymdevgrpname> split -rbcv
```

9. Re-establish the BCV devices in the recovery Symmetrix as mirrors of the standard devices:

```
#symmir -f <recbcvdev_textfile> -sid <recsymid> est
```

10. Re-establish the BCV/R1 devices in the secondary Symmetrix as mirrors of the standard devices.

```
#symmir -g <prisymdevgrpname> est -rdf
```

Refer to the sample script prirefreshrec in the /opt/cmcluster/toolkit/SGSRDF/cascade/Samples directory for examples of how these commands are used. This script is designed to run only on a node that connects to the primary Symmetrix.

**Data Replication in Failover and Failback Scenarios**

This section describes the data replication procedures for various failover and failback scenarios.

**Scenario 1—Primary Site within the Primary Cluster Fails**

When a failure occurs at the primary site, the hosts are down or the whole site is down, the application package is automatically failover to the secondary site within the primary cluster. Until the problems at the primary site are fixed, and data replication is reestablished, there is no
data protection for the package at the secondary site. Depending on the type of failure and how quickly the primary site is back online, data refresh to the recovery site is still needed. This scenario is illustrated in Figure 8-10.

**Figure 8-10  Failure of Primary Site in Primary Cluster**

Establishing Data Replication from the Secondary to the Recovery Site After Failover to the Secondary Site

After failover, the application is running on secondary site and writing I/O to R2 devices. The data is not remotely protected. The procedure to refresh the data from the secondary Symmetrix to the recovery Symmetrix is the same as the one that is done in steady state. But the procedure is now running on a system in the secondary site; therefore, the options on some of the SYMCLI commands are different.

1. Split the BCV devices in the recovery Symmetrix from the mirror group. This is required to preserve an old copy of the data just in case a failure occurs during data synchronization between the secondary Symmetrix and the recovery Symmetrix that may cause data corruption on the R2 devices in the recovery Symmetrix.

   ```bash
   # symmir -f <recbcvdev_textfile> -sid <recsymid> split
   ``

   The `<recbcvdev_textfile>` is a file that contains a list of the standard device and BCV device pair in the recovery Symmetrix.

2. Freeze the application I/O if the application is writing data to the R1 devices in the primary Symmetrix. The method of freezing the I/O is
Cascading Failover in a Continental Cluster

Data Replication Procedures

application dependent.

3. Split the BCV/R1 devices in the secondary Symmetrix from the
   mirror group and start synchronization with the R2 devices in the
   recovery Symmetrix:

   \#symmir -g <secsymdevgrpname> split

4. Resume the application I/O to the R1 devices in the primary
   Symmetrix if needed. The method of resuming the I/O is application
   dependent.

5. Incrementally establish the SRDF volume pairs between the
   secondary Symmetrix and the recovery Symmetrix. The data is
   synchronized from the secondary Symmetrix BCV/R1 devices to the
   recovery Symmetrix R2 devices

   \#symrdf -g <secsymdevgrpname> est -bcv

6. Use the following command to check the data synchronization from
   the BCV/R1 devices to the R2 devices. If the "RDF Pair STATE"
   column shows the state "Synchronized" for all the devices, the copying
   completed.

   \#symrdf -g <secsymdevgrpname> query -bcv

7. Once the copy completes, split the SRDF link between the secondary
   Symmetrix and the recovery Symmetrix:

   \#symrdf -g <secsymdevgrpname> split -bcv

8. Re-establish the BCV devices in the recovery Symmetrix as mirrors of
   the standard devices:

   \#symmir -f <recbcvdev_textfile> -sid <recsymid> est

9. Re-establish the BCV/R1 devices in the secondary Symmetrix as
   mirrors of the standard devices:

   \#symmir -g < secsymdevgrpname > est

Refer to the sample script secrefreshrec in the
/opt/cmcluster/toolkit/SGSRDF/cascade/Samples directory for
example of how these commands are used. This script is designed to run
only on a node that connects to the secondary Symmetrix.

Failback from the Secondary Site to the Primary Site

Once the problems at the primary site have been fixed, the application
can fail back to the primary site. The current RDF pair states of the
Cascading Failover in a Continental Cluster

Data Replication Procedures

package device groups will be “Split,” which is not handled automatically by the package control script. The following steps are required to move the package back to the primary site.

1. Set the RDF pairs between primary Symmetrix and secondary Symmetrix to fail over state. Either use the following command on a system that connects to the secondary Symmetrix:

   `# symrdf -g <secsymdevgrpname> failover -force`

or use the following command on a system that connects to the primary Symmetrix

   `# symrdf -g <prisymdevgrpname> failover -force`

This command can be run while the package application is still running. The device group has already been failed over, so all this command will do is change the RDF pair state from “Split” to “Failed Over.”

2. Now, halt the application package:

   `# cmhaltpkg <package_name>`

3. Split the BCV/R1 in the secondary Symmetrix from the mirror group to save a good copy of the data. From a system that connects to the secondary Symmetrix:

   `# symmir -g <secsymdevgrpname> split`

From a system that connects to the primary Symmetrix:

   `# symmir -g <prisymdevgrpname> split -rdf`

4. Start the application package on the primary site. Use the following command for all hosts in the primary cluster that may run this package:

   `# cmmodpkg -e -n <host_name> <package_name>`

Then run the package with the following command:

   `# cmmodpkg -e <package_name>`

The package will now start up on its primary host. The package control script will change the R1 devices from “Not Ready” status to “WD” (Write Disabled) status, then an RDF failback will be done on the package device group. The failback will synchronize the R1 devices from the R2 devices. Until the synchronization is complete, the package application may run at a lower performance level.
5. Use the following command to check the RDF pair state between the R1 devices in the primary Symmetrix to the R2 devices in the secondary Symmetrix. From a host that connects to the secondary Symmetrix:

```
# symrdf -g <secsymdevgrpname> query
```

From a host that connects to the primary Symmetrix:

```
# symrdf -g <prisymdevgrpname> query
```

If the “RDF Pair STATE” column shows the state “Synchronized” for all the devices, then proceed to the next step.

6. Re-establish the BCV/R1 devices in the secondary Symmetrix as a mirror of the standard device. From a host that connects to the secondary Symmetrix:

```
# symmir -g <secsymdevgrpname> est
```

From a host that connects to the primary Symmetrix:

```
# symmir -g <prisymdevgrpname> est -rd
```

Scenario 2—Secondary Site within the Primary Cluster Fails

When the secondary site fails, or all SRDF links between the primary Symmetrix and the secondary Symmetrix fail, unless domino mode is used, the application running on the primary site is not aware of this failure and continues to run on the primary site. This scenario is illustrated in Figure 8-11.
Cascading Failover in a Continental Cluster

Data Replication Procedures

Without the secondary site, the current configuration doesn’t provide any means to replicate the new data from the primary Symmetrix directly to the recovery Symmetrix. If the secondary site is down for a long time, the data in the recovery Symmetrix is very out-of-date. If the primary site fails during this time, and the recovery takes over, the customer will have to operate on an old copy of the data. Therefore, it’s important to fix and have the secondary site up and running as soon as possible.

When the secondary site is fixed, the SRDF volume pair between the primary Symmetrix and the secondary Symmetrix will be in “Suspended” mode. If the BCV/R1 in the secondary Symmetrix contains a good copy of the data, to protect this data from corruption in case of rolling disaster, these devices must be split from the mirror group before re-establishing the SRDF volume pairs between the primary Symmetrix and the secondary Symmetrix. Use the following steps:

1. Split the BCV/R1 devices in the secondary Symmetrix from the mirror group. From a host that connects to the primary Symmetrix:

   # symmir -g <prisymdevgrpname> split -rdf

   From a host that connects to the secondary Symmetrix:

   # symmir -g <secisymdevgrpname> split

2. Incrementally establish the SRDF volume pairs between the primary Symmetrix and the secondary Symmetrix. From a host that connects to the primary Symmetrix:

   # symrdf -g <prisymdevgrpname> est

   From a host that connects to the secondary Symmetrix:

   # symrdf -g <secsymdevgrpname> est

3. Use the following command to check the state of the SRDF link from the R1 devices in the primary Symmetrix to the R2 devices in the secondary Symmetrix. Make sure that the state is “Synchronized” before proceeding to the next step. From a host that connects to the primary Symmetrix:

   # symrdf -g <prisymdevgrpname> query

   From a host that connects to the secondary Symmetrix:

   # symrdf -g <secsymdevgrpname> query

4. Re-establish the BCV/R1 devices in the secondary Symmetrix as a mirror of the standard device. From a host that connects to the
Cascading Failover in a Continental Cluster

Data Replication Procedures

primary Symmetrix:

```bash
# symmir -g <prismdevgrpname> est -rdf
```

From a host that connects to the secondary Symmetrix:

```bash
# symmir -g <secsymdevgrpname> est
```

Scenario 3—Entire Primary Cluster Fails

In this scenario, the assumption is that both primary site and secondary site fail at the same time or very close to each other. This scenario is illustrated in Figure 8-12.

**Figure 8-12** Failure of Entire Primary Cluster

Failover from Primary Cluster to Recovery Cluster

After reception of the ContinentalClusters alerts and alarm, the administrators at the recovery site follow the prescribed processes and recovery procedures to start the protected applications on the recovery cluster. Note that data corruption may occur in situation where a disaster occurs at the primary cluster while the data refresh from secondary Symmetrix to the recovery Symmetrix is in progress. The data in the R2 devices in the recovery Symmetrix is not usable. The data can be recovered by restoring an old copy of the data from the BCV devices in the recovery Symmetrix. Execute the following commands to restore the data:

1. Restore the data from the BCV to the R2 devices in the recovery
Cascading Failover in a Continental Cluster

Data Replication Procedures

Symmetrix:

```bash
# symmir -g <recsymdevgrpname> -full restore
```

2. Use the following command to check the data restore progress. If the pair state is “Synchronized,” the data restore is complete.

```bash
# symmir -g <recsymdevgrpname> query
```

3. Once the restore completed, split the BCV devices from the mirror group:

```bash
# symmir -g <recsymdevgrpname> split
```

The data in the recovery Symmetrix may not be current but should be consistent. There is no additional procedure needed. The package control script is programmed to handle this case.

After the application is up and running, re-establish the BCV devices as mirrors of the standard devices for an additional copy of the data:

```bash
# symmir -g <recsymdevgrpname> est
```

Failback from the Recovery Cluster to the Primary Site

The current configuration doesn’t support the application failback to the primary site in the primary cluster unless the secondary site in the primary cluster is up and running. The secondary site has to be repaired first. The application can temporarily fail back to the secondary site while the primary site is still down. Before the application can fail back to either the secondary site or the primary site, the current data need to be restored from the recovery Symmetrix to the secondary Symmetrix and the primary Symmetrix.

Data Restore and Failback from the Recovery Cluster to the Secondary Site within the Primary Cluster

This procedure is used in situation where the application fails back and runs on the secondary site while the primary site is still down.

1. Halt the ContinentalClusters monitor package.

2. Halt the ContinentalClusters recovery packages at the recovery site.

3. Split the BCV/R1 from the mirror group if it's not already in the split state

   The following example shows the command if run on a host that connects to the recovery Symmetrix:

   ```bash
   # symmir -g <recsymdevgrpname> est
   ```
Cascading Failover in a Continental Cluster

Data Replication Procedures

```bash
# symmir -f <secbcvdev_textfile> -sid <secsymid> split
```
The `<secbcvdev_textfile>` is a file that contains a list of the standard device and BCV device pair in the secondary Symmetrix.

The following shows the command if run on a host that connects to the secondary Symmetrix:

```bash
# symmir -g <secsymdevgrpname> split
```

4. Re-establish the RDF volume pairs between the recovery Symmetrix and the secondary Symmetrix. Restore the data from the recovery Symmetrix and the secondary Symmetrix. The following shows the command if run on a host that connects to the recovery Symmetrix:

```bash
# symrdf -g <recsymdevgrpname> -full restore
```
The following shows the command if run on a host that connects to the secondary Symmetrix:

```bash
# symrdf -g <secsymdevgrpname> -full restore -bcv
```

5. Use the following command to check the data restore from the R2 devices in the recovery Symmetrix to the BCV/R1 devices in the secondary Symmetrix. If the “RDF Pair STATE” column shows the state “Synchronized” for all the devices, the data restore completed. The following shows the command if run on a host that connects to the recovery Symmetrix:

```bash
# symrdf -g <recsymdevgrpname> query
```
The following shows the command if run on a host that connects to the secondary Symmetrix:

```bash
# symrdf -g <secsymdevgrpname> query -bcv
```

6. Once the copy completes, split the SRDF link between secondary Symmetrix and recovery Symmetrix. The following shows the command if run on a host that connects to the recovery Symmetrix:

```bash
# symrdf -g <recsymdevgrpname> split
```
The following shows the command if run on a host that connects to the secondary Symmetrix:

```bash
# symrdf -g <secsymdevgrpname> split -bcv
```

7. Re-establish the BCV devices in the secondary Symmetrix as mirrors of the standard devices. Restore the data from the BCV/R1 devices to the R2 standard devices. The following shows the command if run on

Cascading Failover in a Continental Cluster

Data Replication Procedures

a host that connects to the recovery Symmetrix:

```
symmir -f <secbcvdev_textfile> -sid <secsymid> -full restore
```

The following shows the command if run on a host that connects to the secondary Symmetrix:

```
symmir -g <secsymdevgrpname> -full restore
```

8. Use the following command to check the data restore from the BCV/R1 devices in the R2 standard devices in the secondary Symmetrix. If the “RDF Pair STATE” column shows the state "Restored" for all the devices, the data restore completed. The following shows the command if run on a host that connects to the recovery Symmetrix:

```
symmir -f <secbcvdev_textfile> -sid <secsymid> query
```

The following shows the command if run on a host that connects to the secondary Symmetrix:

```
symmir -g <secsymdevgrpname> query
```

9. Once the data restore completed, split the BCV/R1 devices from the mirror group. The following shows the command if run on a host that connects to the recovery Symmetrix:

```
symmir -f <secbcvdev_textfile> -sid <secsymid> split
```

The following shows the command if run on a host that connects to the secondary Symmetrix:

```
symmir -g <secsymdevgrpname> split
```

10. Re-establish the BCV/R1 devices as mirrors of the standard devices. The following shows the command if run on a host that connects to the recovery Symmetrix:

```
symmir -f <secbcvdev_textfile> -sid <secsymid> est
```

The following shows the command if run on a host that connects to the secondary Symmetrix:

```
symmir -g <secsymdevgrpname> est
```

11. Since the recovery cluster has a different cluster ID, the data restore from the recovery Symmetrix to the secondary Symmetrix also copies the recovery cluster’s ID to the secondary Symmetrix. Do the following to change the cluster’s ID on each cluster aware volume group in the secondary Symmetrix before starting the application.
Cascading Failover in a Continental Cluster

Data Replication Procedures

12. Start the application package at the secondary site. Issue the following command for all hosts in secondary site that may run this package:

```
## cmmodpkg -e -n <host_name> <package_name>
```

Then issue the following command from any node on the primary cluster:

```
## cmmodpkg -e <package_name>
```

13. Start the ContinentalClusters monitor package.

Refer to the sample script `recrestoresec` in the `/opt/cmcluster/toolkit/SGSRDF/cascade/Samples` directory for the automation of step 3 to step 10. This script is designed to run only on a node that connects to the secondary Symmetrix.

**Data Restore and Failback from the Recovery Site Directly to the Primary Site in the Primary Cluster**

This procedure is used in situation where both the secondary site and the primary site are fixed and up and running. The package application fails back directly from the Recover cluster to the primary site in the primary cluster.

1. Halt the ContinentalClusters monitor package at the recovery site.

2. Halt the ContinentalClusters recovery packages at the recovery site.

3. Split the BCV/R1 from the mirror group if it's not already in the split state. Use the following command if run on a host that connects to the recovery Symmetrix:

```
# symmir -f <secbcvdev_textfile> -sid <secsymid> split
```

The `<secbcvdev_textfile>` is a file that contains a list of the standard device and BCV/R1 device pairs in the secondary Symmetrix.

Use the following command if run on a host that connects to the secondary Symmetrix:

```
# symmir -g <secsymdevgrpname> split
```

4. Re-establish the RDF volume pairs between the recovery Symmetrix
Cascading Failover in a Continental Cluster

Data Replication Procedures

and the secondary Symmetrix. Restore the data from the recovery Symmetrix and the secondary Symmetrix. Use the following command if run on a host that connects to the recovery Symmetrix:

```
# symrdf -g <recsymdevgrpname> -full restore
```

Use the following command from a host that connects to the secondary Symmetrix:

```
# symrdf -g <secsymdevgrpname> -full restore -bcv
```

5. Use the following command to check the data restore from the R2 devices in the recovery Symmetrix to the BCV/R1 devices in the secondary Symmetrix. If the "RDF Pair STATE" column shows the state "Synchronized" for all the devices, the data restore completed. From a host that connects to the recovery Symmetrix:

```
# symrdf -g <recsymdevgrpname> query
```
From a host that connects to the secondary Symmetrix:

```
# symrdf -g <secsymdevgrpname> query -bcv
```

6. Once the restore completes, split the SRDF link between secondary Symmetrix and recovery Symmetrix. From a host that connects to the recovery Symmetrix:

```
# symrdf -g <recsymdevgrpname> split
```
From a host that connects to the secondary Symmetrix:

```
# symrdf -g <secsymdevgrpname> split -bcv
```

7. Re-establish the BCV/R1 devices in the secondary Symmetrix as mirrors of the standard devices. Restore the data from the BCV/R1 devices to the R2 standard devices. From a host that connects to the recovery Symmetrix:

```
# symmir -f <secbcvdev_textfile> -sid <secsymid> -full restore
```
From a host that connects to the secondary Symmetrix:

```
# symmir -g <secsymdevgrpname> -full restore
```

8. Use the following command to check the data restore from the BCV/R1 devices in the R2 standard devices in the secondary Symmetrix. If the "RDF Pair STATE" column shows the state "Restored" for all the devices, the data restore completed. From a host that connects to the recovery Symmetrix:

```
# symmir -f <secbcvdev_textfile> -sid <secsymid> query
```
Cascading Failover in a Continental Cluster

Data Replication Procedures

From a host that connects to the secondary Symmetrix:

```bash
# symmir -g <secsymdevgrpname> query
```

9. Once the data restore completed, split the BCV/R1 devices from the mirror group. From a host that connects to the recovery Symmetrix:

```bash
# symmir -f <secbcvdev_textfile> -sid <secsymid> split
```

From a host that connects to the secondary Symmetrix:

```bash
# symmir -g <secsymdevgrpname> split
```

10. Set the RDF pairs between primary Symmetrix and secondary Symmetrix to fail over state. Run the following command on a host that connects to the secondary Symmetrix:

```bash
# symrdf -g <secsymdevgrpname> failover -force
```

11. Since the recovery cluster has a different cluster ID, the data restore from the recovery Symmetrix to the secondary Symmetrix also copies the recovery cluster’s ID to the secondary Symmetrix. Do the following to change the cluster’s ID on each cluster aware volume group in the secondary Symmetrix before starting the application package:

```bash
# vgchange -c n /dev/<vg_name>
# vgchange -c y /dev/<vg_name>
```

12. Start the package application at the primary site. Issue the following command for all hosts in the primary cluster that may run this package:

```bash
# cmmodpkg -e -n <host_name> <package_name>
```

Start the package with the following command:

```bash
# cmmodpkg -e <package_name>
```

13. Start the ContinentalClusters monitor package on the recovery cluster.

14. Use the following command to check the RDF pair state between the R1 devices in the primary Symmetrix to the R2 devices in the secondary Symmetrix. From a host that connects to the secondary Symmetrix:

```bash
# symrdf -g <secsymdevgrpname> query
```

From a host that connects to the primary Symmetrix:
Cascading Failover in a Continental Cluster

Data Replication Procedures

```bash
# symrdf -g <prisymdevgrpname> query
```

If the “RDF Pair STATE” column shows the state “Synchronized” for all the devices, then proceed to the next step.

15. Re-establish the BCV/R1 devices in the secondary Symmetrix as mirrors of the standard devices. From a host that connects to the secondary Symmetrix:

```bash
# symmir -g <secsymdevgrpname> est
```

From a host that connects to the primary Symmetrix:

```bash
# symmir -g <prisymdevgrpname> est -rdf
```

Refer to the sample script `recrestoresec` in the `/opt/cmcluster/toolkit/SGSRDF/cascade/Samples` directory for the automation of steps 3 to step 10. This script is designed to be run only on a node that connects to the secondary Symmetrix.
Package Control Script
Variables for MetroCluster/CA

This appendix lists all MC/ServiceGuard package control script variables that have been modified or added for MetroCluster with Continuous Access XP. It is recommended that you use the default settings for most of these variables, so exercise caution when modifying them:

AUTO_FENCEDATA_SPLIT (Default=1)
This parameter applies only when the fence level is set to “data”, which will cause the application to fail if the ESCON link fails or if the remote site fails.
Values:
0—Do not start up the package at the primary site. Require user intervention to either fix the hardware problem or to set the FORCEFLAG. Use this value to ensure that the SVOL data is always consistent despite the tradeoff of long application downtime while the ESCON link and/or the remote site are being repaired.
1—(Default) Start up the package at the primary site. Request the local disk array to automatically split itself from the remote array. This will ensure that the application will be able to start up at the primary site without having to fix the hardware problems immediately. When the ESCON link and/or the remote site is repaired, the user must manually use the command "pairresync" to re-join the PVOL and SVOL. Until that command successfully completes, the PVOL will NOT be remotely protected and the SVOL data may be inconsistent. Use this value to minimize the down time of the application with the tradeoff of having to manually resynchronize the pairs while the application is running at the primary site.

AUTO_PFUSSMPL (Default=0)
This AUTO variable is used in asynchronous mode only. It is used to protect any data that is in the side file.
Package Control Script Variables for MetroCluster/CA

on the MCU during failback to the primary side.
The variable is used in situations where all hosts on the primary side have failed, we failed over to the secondary side, the primary site was restored, but the CA link is still suspended due to the side file's reaching threshold, and we want to fail back to the primary side.

Values:

0—(Default) Do not start the package on the PFUS site when the remote site is SMPL, since there is data in the side file on the PFUS site that we do not want to lose. Exit, not allowing restart on another machine.

1—Start the package on the primary site, after rejoining and resynchronizing from the SMPL (old SVOL) site. Then automatically swap the personalities back (old SVOL becomes SVOL, old PVOL becomes PVOL). This option should be chosen only for those organizations that do not have peer data centers whose functions can be swapped or when you do not wish to swap them. The old disaster site returns to the disaster function, and the old production (primary) site becomes the production site again. This option can also be chosen interactively by setting the FORCEFLAG to override when this value is set to 0.

2—Start the package from the PFUS site (which is probably non-current). Warn that the data is unprotected from a disaster and that the data is non-current. This behavior should be chosen only after very careful consideration, because you will be starting from non-current data. Additional manual procedures should be used to re-join the split PVOL/SMPL and resynchronize to the new SVOL.

AUTO_PSUEPSUS (Default=0)

This variable allows you to control the behavior of PVOLs and SVOLs used by a MetroCluster or ContinentalClusters package after a failover to the secondary site where the following are true:

- The local primary site failed, and we failed over to the secondary site.
A hot takeover -S took place on the secondary site.

The primary site is ready for recovery. The package was halted on the secondary site and restarted on the primary site.

The most common state combination to fail back from an outage on the primary site is PVOL_PSUE and SVOL_PSUS. By default, this variable tells MetroCluster or ContinentalClusters to automate most of the recovery process. The user needs to halt the package on the secondary site and restart it on the primary site once all hardware issues have been resolved.

Values:

0—Do not start the package. Exit 1.
1—(Default) Start up the package on the PVOL side and warn the user.

NOTE

In asynchronous mode, any data that was in the side file when the primary site failed will be lost in this process. If you do not want to lose this data, then set AUTO_PSUESUS to 0.

AUTO_PSUESMPL (Default=1)

This variable allows you to control the behavior of PVOLs and SVOLs used by a MetroCluster or ContinentalClusters package in the following cases:

There was a power loss at the primary site followed by a manual setting of the volume to SMPL, then the primary site was restored, and we want to move the package back to the primary site (on purpose or application failure due to disk array problems) without resetting the CA link

or

There was an ESCON error, HDD error, or adapter error (i.e., the SVOL may not be usable)

Values:
Package Control Script Variables for MetroCluster/CA

0—Do not start up the package at the primary site. Assume that the data is more current at the secondary site, and that it may not be possible to resynchronize due to the PSUE condition.

1—(Default) Start up the package. Attempt to resynchronize from the secondary site. Note that this may fail, depending on what caused the PSUE condition. If the application fails to resynchronize due to the PSUE condition, the application can be started without resynchronization by setting the FORCEFLAG (same behavior as value=2).

2—Start up the package and warn that the data is unprotected from a disaster. Do not attempt resynchronization of the data from the secondary site.

AUTO_PSUSSMPL (Default=1)

This variable allows you to control the behavior of PVOLs and SVOLs used by a MetroCluster or ContinentalClusters package when all hosts on the primary site have failed, we failed over to the secondary site, the primary site was restored, and we want to fail back to the primary site.

Values:

0—Do not start the package on the PSUS site when the remote site is SMPL, since the SMPL site probably has more current data. Exit, allowing restart on another machine.

1—(Default) Start the package on the primary site, after rejoining and resynchronizing from the SMPL (old SVOL) site. Then automatically swap the personalities back (old SVOL becomes SVOL, old PVOL becomes PVOL). This option should be chosen only for those organizations that do not have peer data centers whose functions can be swapped or when you do not wish to swap them. The old disaster site returns to the disaster function, and the old production (primary) site becomes the production site again. This option can also be chosen interactively by setting the FORCEFLAG to override when this value is set to 0.
Package Control Script Variables for MetroCluster/CA

2—Start the package from the PSUS site (which is probably non-current). Warn that the data is unprotected from a disaster and that the data is non-current. This behavior should be chosen only after very careful consideration, because you will be starting from non-current data. Additional manual procedures should be used to re-join the split PVOL/SMPL and resynchronize to the new SVOL.

AUTO_SMPLNORMT (Default=1)
This variable allows you to control the behavior of PVOLS and SVOLS used by a MetroCluster or ContinentalClusters package in the following case:
There was a failback to the primary site after it was repaired, but the old PVOL was manually split, and all hosts at the disaster site have failed. This situation is not supported, since it involves a manual state change.
Values:
0—Do not start up the package if there is a local host failure at the disaster site.
1—(Default) Start up the package.

AUTO_SMPLPAIR (Default=2)
This variable allows you to control the behavior of PVOLS and SVOLS used by a MetroCluster or ContinentalClusters package in the following cases:
The SVOL site was manually put into SMPL state, then the primary site has failed, the secondary site has taken over (and now owns the data, which has probably been updated), and the primary site was repaired. We want to later manually swap the primary and the secondary using the procedure documented in Chapter 3.

or

We are restarting for some reason on the secondary side (intra-site failover) and we do not want to do a PVOL / SVOL Swap.
Values:
0—Do not start the package on the SMPL (old SVOL) site when the primary site has been repaired and is back online. Exit, allowing restart on another machine (which may be on the opposite site).

1—Start the package from the SMPL (old SVOL) site, and automatically swap the personalities (old SVOL becomes the PVOL, old PVOL becomes SVOL) and then resynchronize from the old SVOL to the old PVOL. This option should be chosen only for those organizations that have peer data centers whose functions can be swapped, and you wish to swap them. The old disaster site becomes the production (primary) site and the old production (primary) site becomes the new disaster site. This option can also be chosen interactively by setting the FORCEFLAG to override the default value of 2.

2—(Default) Start the package from the simplex side with a warning that the data is unprotected from a disaster.

AUTO_SMPLPSUS  (Default=1)

This variable allows you to control the behavior of PVOLs and SVOLs used by a MetroCluster or ContinentalClusters package when a previous failover occurred to the disaster site followed by repair of the primary site without resynchronization of the old primary site, followed by a disaster site host failure. In this case, the SVOL was manually split off, and the PVOL was either already in PSUS or manually set to PSUS. There was then a local failover on the secondary site.

Values:

0—Do not start up the package, but exit allowing restart on another host. This would eventually result in a failback to the primary site.

1—(Default) Start up the package and warn that the data is unprotected from a disaster. This behavior assumes that we should not fail back to the primary site unless specifically told to do so.
AUTO_SVOLPFUS  (Default=0)

This variable is used in asynchronous mode when the PVOL and SVOL both have the state of suspended (PFUS) due to the side file's reaching threshold. When the PVOL and SVOL are in this state, the data may be consistent, but it may not be current. When starting the package in this state, you run the risk of losing any data that is being tracked on the bitmap of the PVOL side. This variable is only used in asynchronous mode.

Values:
0—(Default) Do not start up the package at the secondary site. User intervention is required to either fix the problem by resynchronizing the PVOL and SVOL or by setting the FORCEFLAG, allowing restart on another node.
1—Start up the package after making the SVOL writable. The risk of using this option is that the SVOL data may actually be inconsistent and the application may fail. However, there is also the chance that the data is actually consistent, and it is therefore okay to start up the application.

AUTO_SVOLPSUE  (Default=0)

This parameter applies when the PVOL and SVOL both have the state of PSUE. This state combination will occur when there is an ESCON link or other hardware failure, or when the SVOL side is in a PSUE state while we cannot communicate with the PVOL side. This will only be possible in asynchronous mode.

The SVOL side will become PSUE after the ESCON link timeout value has been exceeded, at which time the PVOL side will try to flush any data in the side file to the SVOL side. If this flush is unsuccessful, then the data on the SVOL side may not be current.

Values:
0—(Default) Do not start up the package at the secondary site. User intervention is required to either fix the problem by resynchronizing the PVOL and SVOL or by setting the FORCEFLAG, allowing the
package to try another node.

1—Start up the package on the SVOL side. The risk of using this option is that the SVOL data may actually be inconsistent and the application may fail. However, there is also a chance that the data is actually consistent, and it is therefore okay to start up the application.

**AUTO_SVOLPSUS** (Default=0)

This parameter applies when both the PVOL and SVOL have the state of suspended (PSUS). The problem with this situation is that we cannot determine the previous state: COPY or PAIR. If the previous state was PAIR, it is completely safe to start up the package at the remote site. If the previous state was COPY, the data at the SVOL site is likely to be inconsistent.

Values:

0—(Default) Do not start up the package at the secondary site. Require user intervention to either fix the problem by resynchronizing the PVOL and SVOL or by setting the FORCEFLAG.

1—Start up the package after making the SVOL writeable. The risk of using this option is that the SVOL data may actually be inconsistent and the application may fail. However, there is also a chance that the data is actually consistent, and it is therefore okay to startup the application.

**CLUSTER_TYPE**

This parameter defines the clustering environment in which the script is used. Should be set to “metro” if this is a MetroCluster environment and “continental” if this is a ContinentalClusters environment. A type of “metro” is supported only when the HP MetroCluster product is installed. A type of “continental” is supported only when the HP ContinentalClusters product is installed.
Package Control Script Variables for MetroCluster/CA

DC1HOST[ ]
An array of the hosts in the DC1 data center.

DC2HOST[ ]
An array of the hosts in the DC2 data center.

DEVICE_GROUP
Specifies the Raid Manager device group for this package.

FENCE
Fence level. Possible values are NEVER, DATA, and ASYNC. Use ASYNC for improved performance over long distances.

If a Raid Manager device group contains multiple items where either the PVOL or SVOL devices reside on more than a single XP Series array, then the Fence level must be set to “data” in order to prevent the possibility of inconsistent data on the remote side if an ESCON link or an array goes down. The side effect of the “data” fence level is that if the package is running and a link goes down, an array goes down, or the remote data center goes down, then write(1) calls in the package application will fail, causing the package to fail.

HORCMINST
This is the instance of the Raid Manager that the control script will communicate with. This instance of Raid Manager must be started on all nodes before this package can be successfully started. (Note: If this variable is not exported, Raid Manager commands used in this script may fail).

HORCTIMEOUT (Default=360)
This variable is used only in asynchronous mode when the horctakeover command is issued; it is ignored in synchronous mode. The value is used as the timeout value in the horctakeover command, -t <timeout>. The value is the time to wait while horctakeover re-synchronizes the delta data from the PVOL to the SVOL. It is used for swap-takeover and SVOL takeover.
Appendix A

Package Control Script Variables for MetroCluster/CA

If the timeout value is reached and a timeout occurs, horctakeover returns the value EX_EWSTOT. The unit is seconds.

In asynchronous mode, when there is an ESCON link failure, both the PVOL and SVOL sides change to a PSUE state. However, on the SVOL side, this change will not take place until the ESCON link timeout value, configured in the Service Processor (SVP), has been reached. If the horctakeover command is issued during this timeout period, the horctakeover command could fail if its timeout value is less than that of the ESCON link timeout. Therefore, it is important to set the HORCTIMEOUT variable to a value greater than the ESCON link timeout value. The default ESCON link timeout value is 5 minutes (300 seconds). A suggested value for HORCTIMEOUT is 360 seconds.

During package startup, the default startup timeout value of the package is set to NO_TIMEOUT in the package ASCII file. However, if there is a need to set a startup timeout value, then the package startup timeout value must be greater than the HORCTIMEOUT value, which is greater than the ESCON link timeout value:

\[ \text{Pkg Startup Timeout} > \text{HORCTIMEOUT} > \text{ESCON link timeout value} \]

\[ \text{MULTIPLE_PVOL_OR_SVOL_FRAMES_FOR_PKG} \] (Default=0)

This parameter must be set to 1 if a PVOL or an SVOL for this package resides on more than one XP256 or XP 512 frame. Currently, only a value of 0 is supported for this parameter.

**NOTE**

Future releases may allow a value of 1.

Values:

0—(Default) Single frame.

1—Multiple frames. If this parameter is set to 1, then the device group must be created with the “data” fence
level, and the FENCE parameter must be set to “data” in this script.

**PKGDIR**

Contains the full path name of the package directory. This directory must be unique for each package to prevent the status file from being overwritten when there are multiple packages. The operator may create the FORCEFLAG file in this directory.

**WAITTIME**

Seconds to wait for each “pairevtwait” interval. (Note: do not set this to less than 300 seconds because the disks have some long final processing when the copy state reaches 100%).
Package Control Script Variables for MetroCluster/CA
Appendix B

Package Control Script Variables for MetroCluster/SRDF

This appendix lists all MC/ServiceGuard control script variables that have been modified or added for MetroCluster with EMC SRDF. It is recommended that you use the default settings for most of these variables, so exercise caution when modifying them:

**AUTOR1RWNL**  
Default: 0  
This variable indicates that when the package is being started on an R1 host, the Symmetrix is in a Read/Write state, and the SRDF links are down, the package will be automatically started. Although the script cannot check the state of the Symmetrix on the R2 side to validate conditions, the Symmetrix on the R1 side is in a 'normal' state. To require operator intervention before starting the package under these conditions, set AUTOR1RWNL = 1 and create the file /etc/cmcluster/package_name/FORCEFLAG.

**AUTOR1UIP**  
Default: 1  
This variable indicates that when the package is being started on an R1 host and the Symmetrix is being synchronized from the Symmetrix on the R2 side, the package will halt unless the operator creates the $PKGDIR/FORCEFLAG file. The package halts because performance degradation of the application will occur while the resynchronization is in progress. More importantly, it is better to wait for the resynchronization to finish to guarantee that the data are consistent even in the case of a rolling disaster where a second failure occurs before the first failure is recovered from. To always automatically start the package even when resynchronization is in progress, set AUTOR1UIP = 0.
Doing so will result in inconsistent data in case of a rolling disaster.

**AUTOR2RWNL**  
Default: 0  
A value of 0 for this variable indicates that when the package is being started on an R2 host, the Symmetrix is in a read/write state, and the SRDF links are down, the package will be automatically started. Although the script cannot check the state of the Symmetrix on the R1 side to validate conditions, the Symmetrix on the R2 side is in a reasonable state. The default setting allows package failover among nodes on the R2 side. To require operator intervention before starting the package under these conditions, set AUTOR2RWNL=1 and create the file

/etc/cmcluster/package_name/FORCEFLAG.

**AUTOR2WDNL**  
Default: 0  
A value of 0 for this variable indicates that when the package is being started on an R2 host, the Symmetrix is in a Write-disabled state, and the SRDF links are down, the package will be automatically started. Although we cannot check the state of the Symmetrix on the R1 side to validate conditions, the Symmetrix on the R2 side is in a ‘normal’ state. To require operator intervention before starting the package under these conditions, set AUTOR2WDNL=1 and create the file

/etc/cmcluster/package_name/FORCEFLAG.

**AUTOR2XXNL**  
Default: 0  
A value of 0 for this variable indicates that when the package is being started on an R2 host and at least one (but not all) SRDF links are down, the package will be automatically started. This will normally be the case when the ‘Partitioned+Suspended’ RDF Pair state exists. We cannot check the state of all Symmetrix volumes on the R1 side to validate conditions, but the Symmetrix on the R2 side should be in a ‘normal’ state. To require operator intervention before starting the package under these conditions, set AUTOR2XXNL=1.

**CLUSTER_TYPE**  
This parameter defines the clustering environment in which the script is used. Should be set to “metro” if this
is a MetroCluster environment and “continental” if this is a ContinentalClusters environment. A type of “continental” is supported only when the HP ContinentalClusters product is installed.

**CONSISTENCYGROUPS**  Default: 0

This parameter tells MetroCluster whether or not consistency groups were used in configuring the R1 and R2 volumes on the Symmetrix frames. A value of 0 is the normal setting if you are not using consistency groups. A value of 1 indicates that you are using consistency groups. (Consistency groups are required for M by N configurations.)

**PATH**

Has been modified to include the path name of the Symmetrix SymCLI commands. This should be set to the default location of `/usr/symcli/bin` unless you have changed the location.

**PKGDIR**

In addition to indicating the location of the package control scripts, this variable also will be used for the following files:

- `symcfg.out` - contains the results of `symcfg list` command, used for model and revision information.
- `symrdf.out` - contains the results of `symrdf query` commands run from the package control script.
- `awk.out` - contains the output from using `awk` to parse the `symrdf.out` file.
- `FORCEFLAG` - forces a package to start automatically under certain circumstances if this file is present. The `cmrunpkg packagename` command deletes this file, so it must be recreated each time if you wish to start packages under these circumstances.

**PKGR1SYMGRP**

This variable contains the name of the Symmetrix device group for the package on the R1 side in a 1 by 1 configuration. In an M by N configuration, it contains the name of the consistency group for the package on the R1 side.

**PKGR2SYMGRP**

This variable contains the names of the Symmetrix device group for the package on the R2 side in a 1 by 1
configuration. In an M by N configuration, it contains the name of the consistency group for the package on the R2 side.

**R1HOST[n]** This array contains the names of all package hosts on the R1 side. Array indices must start at [0] and be incremented consecutively. The variables R1HOST and R2HOST must be customized differently for each package. The order of the hosts in the array should be the same as the order of the hosts in the package ASCII file for the parameter NODE_NAME.

**R2HOST[n]** This array contains the names of all package hosts on the R2 side. Array indices must start at [0] and be incremented consecutively. The variables R1HOST and R2HOST must be customized differently for each package. The preferred host for the package should be on the R1 side and should be listed first. The order of the hosts in the array should be the same as the order of the hosts in the package ASCII file for the parameter NODE_NAME.

**RETRY** Default: 5.

This is the number of times a SymCLI command is repeated before returning an error. Use the default value for the first package, and slightly larger numbers for additional packages making sure that the total of RETRY * RETRYTIME is approximately 5 minutes.

Larger values for RETRY may cause the start-up time for the package to increase when there are multiple packages starting concurrently in the cluster that access the Symmetrix arrays.

**RETRYTIME** Default: 5.

This is the is the number of seconds between retries. The default value of 5 seconds should be used for the first package. The values should be slightly different for other packages. RETRYTIME should increase by two seconds for each package. The product of RETRY * RETRYTIME should be approximately five minutes. These variables are used to decide how often and how many times to retry the Symmetrix status and state change commands.
Package Control Script Variables for MetroCluster/SRDF

Larger values for `RETRYTIME` may cause the start-up time for the package to increase when there are multiple packages starting concurrently in the cluster that access the Symmetrix arrays.
Package Control Script Variables for MetroCluster/SRDF
C Configuration File Parameters for ContinentalClusters

This appendix lists all ContinentalClusters configuration file variables. See Chapter 5, “Building a Continental Cluster,” for suggestions on coding these parameters.

CLUSTER_ALARM [Minutes] MINUTES [Seconds] SECONDS

This is a time interval, in minutes and/or seconds, after which the notifications defined in the associated NOTIFICATION parameters are sent and failover to the Recovery Cluster using the cmrecovercl command is enabled. This number must be a positive integer. Minimum is 30 seconds, maximum is 3600 seconds or 60 minutes (one hour).

CLUSTER_ALERT [Minutes] MINUTES [Seconds] SECONDS

This is a time interval, in minutes and/or seconds, after which the notifications defined in the associated NOTIFICATION parameters are sent. Failover to the Recovery Cluster using the cmrecovercl command is not enabled at this time. This number must be a positive integer. Minimum is 30 seconds, maximum is 3600 seconds or 60 minutes (one hour).

CLUSTER_DOMAIN domainname

This is the domain of the nodes in the previously specified cluster. This domain is appended to the NODE_NAME to provide a full system address across the WAN.

CLUSTER_EVENT Clustername/Status

This is a cluster name associated with one of the following changes of status:

- up - the cluster is up and running
- unreachable - the cluster is unreachable
- down - the cluster is down, but nodes are responding
Configuration File Parameters for ContinentalClusters

- **error** - an error is detected
  The maximum length is 47 characters.
  When the MONITORING_CLUSTER detects a change in status, one or more notifications are sent, as defined by the NOTIFICATION parameter, at time intervals defined by the CLUSTER_ALERT and CLUSTER_ALARM parameters.

**CLUSTER_NAME** *clustername*

The name of a member cluster within the continental cluster. It should be the same name that is defined in the MC/ServiceGuard cluster configuration ASCII file. Maximum size is 40 characters.

All nodes in the cluster should be listed after this variable using the NODE_NAME variable. A MONITOR_PACKAGE_NAME and MONITOR_INTERVAL should also be associated with each CLUSTER_NAME.

**CONTINENTAL_CLUSTER_NAME** *name*

The name of a continental cluster managed by the ContinentalClusters product. Maximum size is 40 characters. This name cannot be changed after the configuration is applied. You must first delete the existing configuration if you want to choose a different name.

**DATA_RECEIVER_PACKAGE** *clustername*/*packagename*

This variable is only used if the data replication is carried out by a separate software application that must be kept highly available. If the replication software uses a receiver process, you include this variable in the configuration file. Maximum size is 81 characters.

The parameter consists of a pair of names: the name of the cluster that receives the data to be replicated (usually the Recovery Cluster) as defined in the MC/ServiceGuard cluster configuration ASCII file, followed by a slash (/), followed by the name of the data replication receiver package as defined in the MC/ServiceGuard package configuration ASCII file.
Configuration File Parameters for ContinentalClusters

Some replication software may only have a receiver package as separate package because the sender package is built into the application.

DATA_SENDER_PACKAGE  clustername/packagename

This variable is only used if the data replication is carried out by a separate software application that must be kept highly available. If the replication software uses a sender process, you include this variable in the configuration file. Maximum size is 81 characters.

The parameter consists of a pair of names: the name of the cluster that sends the data to be replicated (usually the Primary Cluster) as defined in the MC/ServiceGuard cluster configuration ASCII file, followed by a slash ("/"), followed by the name of the data replication sender package as defined in the MC/ServiceGuard package configuration ASCII file. Some replication software may only have a receiver package as separate package because the sender package is built into the application.

MONITOR_INTERVAL  n

The interval, in seconds, that the ContinentalClusters monitor polls the cluster, nodes, and packages to see if the status has changed. This number must be an integer. The minimum value is 30 seconds, the default is 60 seconds, and the maximum is 300 seconds (5 minutes).

MONITOR_PACKAGE_NAME packagename

This is the name of the MC/ServiceGuard package containing the ContinentalClusters monitor. Maximum size is 40 characters.

MONITORING_CLUSTER  Name

This is name of the cluster that polls the cluster named in the CLUSTER_EVENT and sends notification. Maximum length is 40 characters.

NODE_NAME nodename

This is the unqualified node name as defined in the
DNS name server configuration. Maximum size is 40 characters.

NOTIFICATION Destination “Message”

This is a destination and message associated with a specific CLUSTER_ALERT or CLUSTER_ALARM. The maximum size of the message string is 170 characters including the quotation marks. The message string must be entered on a separate single line in the configuration file.

The following destinations are acceptable:

- **CONSOLE** - write the specified message to the console.
- **EMAIL Address** - send the specified message to an email address. You can use an email address provided by a paging service to set up automatic paging. Consult your pager service provider for details.
- **OPC Level** - send the specified message to OpenView IT/Operations. The Level may be 8 (normal), 16 (warning), 64 (minor), 128 (major), or 32 (critical).
- **SNMP Level** - send the specified message as an SNMP trap. The Level may be 1 (normal), 2 (warning), 3 (minor), 4 (major), or 5 (critical).
- **SYSLOG** - Append a notice of the specified message to the /var/adm/syslog/syslog.log file. Note that the text of the message is not placed in the syslog file, only a notice from the monitor.
- **TCP Nodename:Portnumber** - send the specified message to a TCP port on the specified node.
- **TEXTLOG Pathname** - append the specified message to a specified text log file.
- **UDP Nodename:Portnumber** - send the specified message to a UDP port on the specified node.

Any number of notifications may be associated with a given alert or alarm.
Configuration File Parameters for ContinentalClusters

PRIMARY_PACKAGE  ClusternamelPackagename

This is a pair of names: the name of a cluster as defined in the MC/ServiceGuard cluster configuration ASCII file, followed by a slash ("/"), followed by the name of the primary package as defined in the MC/ServiceGuard package configuration ASCII file. Maximum size is 81 characters.

RECOVERY_GROUP_NAME  name

This is a name for the set of related primary packages on one cluster and the recovery packages on another cluster that protect the primary packages. The maximum size is 40 characters.

You create a recovery group for each package that should be started on the recovery cluster in case of a failure on the primary cluster. A PRIMARY_PACKAGE and RECOVERY_PACKAGE should be associated with each RECOVERY_GROUP_NAME.

RECOVERY_PACKAGE  ClusternamelPackagename

This is a pair of names: the name of the recovery cluster as defined in the MC/ServiceGuard cluster configuration ASCII file, followed by a slash ("/"), followed by the name of the recovery package as defined in the MC/ServiceGuard package configuration ASCII file. Maximum size is 81 characters.
Configuration File Parameters for ContinentalClusters
This appendix lists all commands and daemons used with ContinentalClusters. Manual pages are also available online.

```
cmapplyconcl [-v] [-C] filename
```

This command verifies the ContinentalClusters configuration as specified in `filename`, creates or updates the binary, and distributes it to all nodes in the continental cluster. It is not necessary to halt the ServiceGuard cluster in order to run this command; however, the ContinentalClusters monitor package must be halted.

If `cmapplyconcl` is specified when the continental cluster has already been configured, the configuration will be updated with the configuration changes.

The `cmapplyconcl` command must be run when a configuration change is made to the ServiceGuard cluster that impacts the ContinentalClusters configuration. For example, if a node is added to the ServiceGuard cluster, the ContinentalClusters ASCII file should be edited to include the new `NODE_NAME`.

All nodes within the ServiceGuard cluster must be running prior to the `cmapplyconcl` command being run.

Options are:

```
-v   Verbose mode displays all messages.
-C filename   The name of the ASCII configuration file. This is a required parameter.
```

```
cmcheckconcl [-v] -C filename
```

This command verifies the ContinentalClusters configuration specified in `filename`. It is not necessary to halt the ServiceGuard cluster in order to run this command.
ContinentalClusters Command and Daemon Reference

command; however, the ContinentalClusters monitor package must be halted. This command will parse the ASCII_file to ensure proper syntax, check parameter lengths, and validate object names such as the CLUSTER_NAME and NODE_NAME. Options are:

- \texttt{C \textit{filename}} The name of the ASCII configuration file. This is a required parameter.

\texttt{cmclrmond}

This is the ContinentalClusters monitor daemon that provides notification of remote cluster status through the Event Monitoring Service (EMS). This monitor runs on both the primary and recovery clusters. The \texttt{cmclsentryd} daemon notifies \texttt{cmclrmond} of any change in cluster status. Log messages are written to the EMS log file \texttt{/etc/resmon/log/api.log} on the node where the monitor was running when it detected a status event.

\texttt{cmclsentryd}

This daemon, which is run from the monitor package (ccmonpkg) starts up the ContinentalClusters monitor \texttt{cmclrmond}. Messages are logged to log file \texttt{/var/adm/cmconcl/sentryd.log}, which may be read using the \texttt{cmreadlog} command.

\texttt{cmdeleteconcl [-f]}

This command is used to delete the ContinentalCluster configuration from the entire ContinentalCluster.

Options are:

- \texttt{-f} Delete the configuration files on all reachable nodes without further prompting. If this option is not used and if some nodes are unreachable, you will be prompted to indicate whether to proceed with deleting the configuration on the reachable nodes. If this option is used and some node has configuration files for a continental cluster with a different name, you will be prompted to
indicate whether to proceed with deleting the configuration on that node.

**cmcmd**

This daemon is the Object Manager, which communicates with ServiceGuard to provide information about cluster objects to the ContinentalClusters monitor. Messages are logged to log file /var/opt/cmom/cmomd.log, which may be read using the cmreadlog command.

**cmqueryconcl filename**

This command cmqueryconcl creates a template ASCII ContinentalClusters configuration file. The ASCII file should be customized for a specific ContinentalClusters environment. After customization, this file should be verified by the cmcheckconcl command and distributed by using the cmapplyconcl command. If an ASCII file is not provided, output will be directed to stdout.

This command should be run as the first step in preparing for ContinentalClusters configuration.

Options are:

- **-v**  
  Verbose mode displays all messages.

- **-C filename**  
  Declares an alternate location for the configuration file. The default is /etc/cmcluster/cmoncl.config.

**cmreadlog -f input_file [output_file]**

This command formats the content of Object Manager and other log files for easier reading. The command is used when reading the /var/opt/cmom/cmomd.log file and the /var/adm/cmconcl/cmclsentryd.log file.

Options are:

- **-f input_file**  
  Specifies the name of the managed object file (MOF file) to be read. This is a required parameter.

- **output_file**  
  The name of a file to which the formatted output is written. If no file
is specified, output is written to stdout.

cmrecovercl [-f]

This command performs the recovery actions necessary to start the recovery groups on the current cluster. Care should be taken before issuing this command. It is important to contact the primary cluster site to determine if recovery is necessary prior to running this command.

This command can be issued from any node on the recovery cluster. This command first connects to the ContinentalClusters monitoring package running on the recovery cluster. This may be a different cluster node than where the cmrecovercl command is being run. cmrecovercl connects to the monitoring package to verify that the primary cluster is in an Unreachable or Down state. If the primary cluster is reachable and the cluster is Up, this command will fail. Next, the data receiver packages on the recovery cluster (if any) are halted sequentially. Finally, the recovery packages are started on the recovery cluster. The recovery packages are started by enabling package switching globally (cmmodpkg -e) for each package. This will cause the package to be started on the first available node within the recovery cluster.

The cmrecovercl command can only be run on a recovery cluster. The cmrecovercl command will fail if there has not been sufficient time since the primary cluster became unreachable. This command is only enabled after the time as configured via CLUSTER_ALARM parameters has been reached. Once a cluster alarm has been triggered, this command will be enabled and can be run. -f option can be used to enable the command after the time as configured via CLUSTER_ALERT parameters has been reached.

Options are:

- **-f**

  The force option enables cmrecovercl to function even though a CLUSTER_ALARM has not been
cmviewconcl [-v]

This command allows you to view the status and much of the configuration of a continental cluster.

This command should be run as the last step when creating a ContinentalClusters configuration to confirm the cluster status, or any time you would like to know cluster status.

Options are:

- -v                      Verbose mode displays all messages.
Glossary

A

application restart Starting an application, usually on another node, after a failure. Application can be restarted manually, which may be necessary if data must be restarted before the application can run (example: Business Recovery Services work like this.) Applications can be restarted by an operator using a script, which can reduce human error. Or applications can be started on the local or remote site automatically after detecting the failure of the primary site.

arbitrator Nodes in a disaster tolerant architecture that act as tie-breakers in case all of the nodes in a data center go down at the same time. These nodes are full members of the MC/ServiceGuard cluster and must conform to the minimum requirements. The arbitrator must be located in a third data center to ensure that the failure of an entire data center does not bring the entire cluster down. See also quorum server.

asynchronous data replication Local I/O will complete without waiting for the replicated I/O to complete; however, it is expected that asynchronous data replication will process the I/Os in the original order.

asymmetrical cluster A cluster that has more nodes at one site than at another. For example, an asymmetrical metropolitan cluster may have two nodes in one building, and three nodes in another building. Asymmetrical clusters are not supported in all disaster tolerant architectures.

automatic failover Failover directed by automation scripts or software (such as MC/ServiceGuard) and requiring no human intervention. In a ContinentalClusters environment, the start-up of package recovery groups on the Recovery Cluster without intervention. See also application restart.

B

bi-directional configuration A continental cluster configuration in which each cluster serves the roles of primary and recovery cluster for different recovery groups. Also known as a mutual recovery configuration.
**BC** (Business Copy) A PVOL or SVOL in an HP SureStore XP series disk array that can be split from or merged into a normal PVOL or SVOL. It is often used to create a snapshot of the data taken at a known point in time. Although this copy, when split, is often consistent, it is not usually current.

**BCV** (Business Continuity Volume) An EMC Symmetrix term that refers to a logical device on the EMC Symmetrix that may be merged into or split from a regular R1 or R2 logical device. It is often used to create a snapshot of the data taken at a known point in time. Although this copy, when split, is often consistent, it is not usually current.

**Business Recovery Service** Service provided by a vendor to host the backup systems needed to run mission critical applications following a disaster.

**C**

**campus cluster** A single cluster that is geographically dispersed within the confines of an area owned or leased by the organization such that it has the right to run cables above or below ground between buildings in the campus. Campus clusters are usually spread out in different rooms in a single building, or in different adjacent or nearby buildings.

**cascading failover** Cascading failover is the ability of an application to fail from a primary to a secondary location, and then to fail to a recovery location on a different site. The primary location contains a metropolitan cluster built with MetroCluster EMC SRDF, and the recovery location has a standard MC/ServiceGuard cluster.

**client reconnect** Users access to the backup site after failover. Client reconnect can be transparent, where the user is automatically connected to the application running on the remote site, or manual, where the user selects a site to connect to.

**cluster** An MC/ServiceGuard cluster is a networked grouping of HP 9000 series 800 servers (host systems known as nodes) having sufficient redundancy of software and hardware that a single failure will not significantly disrupt service. MC/ServiceGuard software monitors the health of nodes, networks, application services, EMS resources, and makes failover decisions based on where the application is able to run.
successfully.

**cluster alarm** Time at which a message is sent indicating that the Primary Cluster is probably in need of recovery. The `cmrecovercl` command is enabled at this time.

**cluster alert** Time at which a message is sent indicating a problem with the cluster.

**cluster event** A cluster condition that occurs when the cluster goes down or enters an **UNKNOWN** state, or when the monitor software returns an error. This event may cause an alert messages to be sent out, or it may cause an alarm condition to be set, which allows the administrator on the Recovery Cluster to issue the `cmrecovercl` command. The return of the cluster to the **UP** state results in a cancellation of the event, which may be accompanied by a cancel event notice. In addition, the cancellation disables the use of the `cmrecovercl` command.

**cluster quorum** A dynamically calculated majority used to determine whether any grouping of nodes is sufficient to start or run the cluster. Cluster quorums prevent split-brain syndrome which can lead to data corruption or inconsistency. Currently at least 50% of the nodes plus a tie-breaker are required for a quorum. If no tie-breaker is configured, then greater than 50% of the nodes is required to start and run a cluster.

**command device** A disk area in the HP SureStore XP series disk array used for internal system communication. You create two command devices on each array, each with alternate links (PV links).

**consistency group** A set of Symmetrix RDF devices that are configured to act in unison to maintain the integrity of a database. Consistency groups allow you to configure R1/R2 devices on multiple Symmetrix frames in MetroCluster/SRDF.

**continental cluster** A group of clusters that use routed networks and/or common carrier networks for data replication and cluster communication to support package failover between separate clusters in different data centers. Continental clusters are often located in different cities or different countries and can span 100s or 1000s of kilometers.

**Continuous Access** A facility provided by the Continuous Access software option available with the HP SureStore E Disk Array XP series.
This facility enables physical data replication between XP series disk arrays.

**data center** A physically proximate collection of nodes and disks, usually all in one room.

**data consistency** Whether data are logically correct and immediately usable; the validity of the data after the last write. Inconsistent data, if not recoverable to a consistent state, is corrupt.

**data currency** Whether the data contain the most recent transactions, and/or whether the replica database has all of the committed transactions that the primary database contains; speed of data replication may cause the replica to lag behind the primary copy, and compromise data currency.

**data loss** The inability to take action to recover data. Data loss can be the result of transactions being copied that were lost when a failure occurred, non-committed transactions that were rolled back as part of a recovery process, data in the process of being replicated that never made it to the replica because of a failure, transactions that were committed after the last tape backup when a failure occurred that required a reload from the last tape backup. **transaction processing monitors** (TPM), message queuing software, and synchronous data replication are measures that can protect against data loss.

**data mirroring** See **mirroring**.

**data recoverability** The ability to take action that results in data consistency, for example database rollback/roll forward recovery.

**data replication** The scheme by which data is copied from one site to another for disaster tolerance. Data replication can be either physical (see **physical data replication**) or logical (see **logical data replication**). In a ContinentalClusters environment, the process by which data that is used by the Primary Cluster packages is transferred to the Recovery Cluster and made available for use on the Recovery Cluster in the event of a recovery.

**database replication** A software-based logical data replication scheme
that is offered by most database vendors.

**disaster** An event causing the failure of multiple components or entire data centers that render unavailable all services at a single location; these include natural disasters such as earthquake, fire, or flood, acts of terrorism or sabotage, large-scale power outages.

**disaster protection** (Don’t use this term?) Processes, tools, hardware, and software that provide protection in the event of an extreme occurrence that causes application downtime such that the application can be restarted at a different location within a fixed period of time.

**disaster recovery** The process of restoring access to applications and data after a disaster. Disaster recovery can be manual, meaning human intervention is required, or it can be automated, requiring little or no human intervention.

**disaster recovery services** Services and products offered by companies that provide the hardware, software, processes, and people necessary to recover from a disaster.

**disaster tolerant** The characteristic of being able to recover quickly from a disaster. Components of disaster tolerance include redundant hardware, data replication, geographic dispersion, partial or complete recovery automation, and well-defined recovery procedures.

**disaster tolerant architecture** A cluster architecture that protects against multiple points of failure or a single catastrophic failure that affects many components by locating parts of the cluster at a remote site and by providing data replication to the remote site. Other components of disaster tolerant architecture include redundant links, either for networking or data replication, that are installed along different routes, and automation of most or all of the recovery process.

**E, F**

**ESCON** Enterprise Storage Connect. A type of fiber-optic channel used for inter-frame communication between EMC Symmetrix frames using EMC SRDF or between HP SureStore E XP series disk array units using Continuous Access XP.

**event log** The default location (/etc/cmcluster/cmconcl/eventlog)
where events are logged on the monitoring ContinentalClusters system. All events are written to this log, as well as all notifications that are sent elsewhere.

**failback** Failing back from a backup node, which may or may not be remote, to the primary node that the application normally runs on.

**failover** The transfer of control of an application or service from one node to another node after a failure. Failover can be manual, requiring human intervention, or automated, requiring little or no human intervention.

**filesystem replication** The process of replicating filesystem changes from one node to another.

**G**

**gatekeeper** A small EMC Symmetrix device configured to function as a lock during certain state change operations.

**H, I**

**heartbeat network** A network that provides reliable communication among nodes in a cluster, including the transmission of heartbeat messages, signals from each functioning node, which are central to the operation of the cluster, and which determine the health of the nodes in the cluster.

**high availability** A combination of technology, processes, and support partnerships that provide greater application or system availability.

**J, K, L**

**local cluster** A cluster located in a single data center. This type of cluster is not disaster tolerant.

**local failover** Failover on the same node; this most often applied to hardware failover, for example local LAN failover is switching to the secondary LAN card on the same node after the primary LAN card has failed.

**logical data replication** A type of on-line data replication that
replicates logical transactions that change either the filesystem or the database. Complex transactions may result in the modification of many diverse physical blocks on the disk.

**LUN** (Logical Unit Number) A SCSI term that refers to a logical disk device composed of one or more physical disk mechanisms, typically configured into a RAID level.

**M**

**M by N** A type of Symmetrix grouping in which up to two Symmetrix frames may be configured on either side of a data replication link in a MetroCluster/SRDF configuration. M by N configurations include 1 by 2, 2 by 1, and 2 by 2.

**manual failover** Failover requiring human intervention to start an application or service on another node.

**MetroCluster** A Hewlett-Packard product that allows a customer to configure an MC/ServiceGuard cluster as a disaster tolerant metropolitan cluster.

**metropolitan cluster** A cluster that is geographically dispersed within the confines of a metropolitan area requiring right-of-way to lay cable for redundant network and data replication components.

**mirrored data** Data that is copied using mirroring.

**mirroring** Disk mirroring hardware or software, such as MirrorDisk/UX. Some mirroring methods may allow splitting and merging.

**mission critical application** Hardware, software, processes and support services that must meet the uptime requirements of an organization. Examples of mission critical application that must be able to survive regional disasters include financial trading services, e-business operations, 911 phone service, and patient record databases.

**mission critical solution** The architecture and processes that provide the required uptime for mission critical applications.

**multiple points of failure (MPOF)** More than one point of failure that can bring down an MC/ServiceGuard cluster.
**multiple system high availability** Cluster technology and architecture that increases the level of availability by grouping systems into a cooperative failover design.

**mutual recovery configuration** A continental cluster configuration in which each cluster serves the roles of primary and recovery cluster for different recovery groups. Also known as a **bi-directional configuration**.

**N**

**network failover** The ability to restore a network connection after a failure in network hardware when there are redundant network links to the same IP subnet.

**notification** A message that is sent following a cluster or package event.

**O**

**off-line data replication**. Data replication by storing data off-line, usually a backup tape or disk stored in a safe location; this method is best for applications that can accept a 24-hour recovery time.

**on-line data replication** Data replication by copying to another location that is immediately accessible. On-line data replication is usually done by transmitting data over a link in real time or with a slight delay to a remote site; this method is best for applications requiring quick recovery (within a few hours or minutes).

**P**

**package alert** Time at which a message is sent indicating a problem with a package.

**package event** A package condition such as a failure that causes a notification message to be sent. Package events can be accompanied by alerts, but not alarms. Messages are for information only; the **cmrecovercl** command is not enabled for a package event.

**package recovery group** A set of one or more packages with a mapping between their instances on the Primary Cluster and their instances on the Recovery Cluster.
physical data replication An on-line data replication method that duplicates I/O writes to another disk on a physical block basis. Physical replication can be hardware-based where data is replicated between disks over a dedicated link (e.g. EMC’s Symmetrix Remote Data Facility or the HP SureStore E Disk Array XP Series Continuous Access), or software-based where data is replicated on multiple disks using dedicated software on the primary node (e.g. MirrorDisk/UX).

planned downtime An anticipated period of time when nodes are taken down for hardware maintenance, software maintenance (OS and application), backup, reorganization, upgrades (software or hardware), etc.

PowerPath A host-based software product from Symmetrix that delivers intelligent I/O path management. PowerPath is required for M by N Symmetrix configurations using MetroCluster/SRDF.

Primary Cluster A cluster in production that has packages protected by the HP ContinentalClusters product.

primary package The package that normally runs on the Primary Cluster in a production environment.

pushbutton failover Use of the cmrecovercl command to allow all package recovery groups to start up on the Recovery Cluster following a significant cluster event on the Primary Cluster.

PV links A method of LVM configuration that allows you to provide redundant disk interfaces and buses to disk arrays, thereby protecting against single points of failure in disk cards and cables.

PVOL A primary volume configured in an XP series disk array that uses Continuous Access. PVOLs are the primary copies in physical data replication with Continuous Access on the XP.

Q

quorum See cluster quorum.

quorum server A cluster node that acts as a tie-breaker in a disaster tolerant architecture in case all of the nodes in a data center go down at the same time. See also arbitrator.
R

R1 The Symmetrix term indicating the data copy that is the primary copy.

R2 The Symmetrix term indicating the remote data copy that is the secondary copy. It is normally read-only by the nodes at the remote site.

Recovery Cluster A cluster on which recovery of a package takes place following a failure on the Primary Cluster.

recovery group failover A failover of a package recovery group from one cluster to another.

recovery package The package that takes over on the Recovery Cluster in the event of a failure on the Primary Cluster.

regional disaster A disaster, such as an earthquake or hurricane, that affects a large region. Local, campus, and proximate metropolitan clusters are less likely to protect from regional disasters.

remote failover Failover to a node at another data center or remote location.

resynchronization The process of making the data between two sites consistent and current once systems are restored following a failure. Also called data resynchronization.

rolling disaster A second disaster that occurs before recovering from a previous disaster, e.g. while data is being synchronized between two data centers after a disaster, one of the data centers fails, interrupting the data synchronization process. Rolling disasters may result in data corruption that requires a reload from tape backups.

S

single point of failure (SPOF) A component of a cluster or node that, if it fails, affects access to applications or services. See also multiple points of failure.

single system high availability Hardware design that results in a single system that has availability higher than normal. Hardware design
examples are:

• n+1 fans
• n+1 power supplies
• multiple power cords
• on-line addition or replacement of I/O cards, memory, etc.

**special device file** The device file name that the HP-UX operating system gives to a single connection to a node, in the format /dev/devtype/filename.

**split-brain syndrome** When a cluster reforms with equal numbers of nodes at each site, and each half of the cluster thinks it is the authority and starts up the same set of applications, and tries to modify the same data, resulting in data corruption. MC/ServiceGuard architecture prevents split-brain syndrome in all cases unless dual cluster locks are used.

**SRDF** (Symmetrix Remote Data Facility) A level 1-3 protocol used for physical data replication between EMC Symmetrix disk arrays.

**SVOL** A secondary volume configured in an XP series disk array that uses Continuous Access. SVOLs are the secondary copies in physical data replication with Continuous Access on the XP.

**SymCLI** The Symmetrix command line interface used to configure and manage EMC Symmetrix disk arrays.

**Symmetrix device number** The unique device number that identifies an EMC logical volume.

**synchronous data replication** Each data replication I/O waits for the preceding I/O to complete before beginning another replication. Minimizes the chance of inconsistent or corrupt data in the event of a rolling disaster.

**T**

**transaction processing monitor (TPM)** Software that allows you to modify an application to store in-flight transactions in an external
location until that transaction has been committed to all possible copies of the database or filesystem, thus ensuring completion of all copied transactions. A TPM protects against data loss at the expense of the CPU overhead involved in applying the transaction in each database replica.

Software that provides a reliable mechanism to ensure that all transactions are successfully committed. A TPM may also provide load balancing among nodes.

**transparent failover** A client application that automatically reconnects to a new server without the user taking any action.

**transparent IP failover** Moving the IP address from one network interface card (NIC), in the same node or another node, to another NIC that is attached to the same IP subnet so that users or applications may always specify the same IP name/address whenever they connect, even after a failure.

**U-Z**

**volume group** In LVM, a set of physical volumes such that logical volumes can be defined within the volume group for user access. A volume group can be activated by only one node at a time unless you are using ServiceGuard OPS Edition. MC/ServiceGuard can activate a volume group when it starts a package. A given disk can belong to only one volume group. A logical volume can belong to only one volume group.

**WAN data replication solutions** Data replication that functions over leased or switched lines. See also continental cluster.
A
  adding a node to ContinentalClusters configuration, 194
  adding a recovery group in ContinentalClusters, 195, 196
  alarms and cluster events, 138
  how used, 139
  alerts and cluster events, 138
  how used, 139
  application recovery in a continental cluster, 134
  applying the continental clusters configuration, 176
  arbitrator nodes, 54, 56, 57, 91, 93
  asynchronous data replication, 27
  AUTO_FENCEDATA_SPLIT in MetroCluster/CA, 299
  AUTO_PFUSSMPL in MetroCluster/CA, 299
  AUTO_PSUEPSUS in MetroCluster/CA, 300
  AUTO_PSUESMPL in MetroCluster/CA, 301
  AUTO_PSUSSMPL in MetroCluster/CA, 302
  AUTO_RUN (PKG_SWITCHING_ENAB LED)
    setting to NO in a continental cluster, 155
  AUTO_SMLNORMT in MetroCluster/CA, 303
  AUTO_SMLPAIR in MetroCluster/CA, 303
  AUTO_SMLPSUS in MetroCluster/CA, 304
  AUTO_SVOLPFUS in MetroCluster/CA, 305
  AUTO_SVOLSUE in MetroCluster/CA, 305
  AUTO_SVOLSUS in MetroCluster/CA, 306
  AUTOR1RWNL in MetroCluster/SRDF, 311
  AUTOR1UIP in MetroCluster/SRDF, 311
  AUTOR2RWNL in MetroCluster/SRDF, 312
  AUTOR2WDNL in MetroCluster/SRDF, 312
  AUTOR2XXNL in MetroCluster/SRDF, 312
  AutoTrespass function, 46
  B
  bi-directional continental cluster, 133
  building a continental cluster, 131
  C
  campus cluster, 53, 90
  definition, 18
  disaster tolerant architecture guidelines, 44
  using FibreChannel, 43
  Cascading, 265
  cascading failover, 265
  continental cluster configuration steps, 280
  data replication procedures, 281
  data storage setup, 269
  primary cluster packages, 278
  recovery cluster packages, 279
  checking the continental clusters configuration, 176
  cluster data entering in ContinentalClusters configuration file, 164
  cluster events in ContinentalClusters, 136
  cluster lock disk with MetroCluster, 92
  cluster maintenance, 40
  CLUSTER_ALARM
    ContinentalClusters configuration file parameter, 317
  CLUSTER_ALERT
    ContinentalClusters configuration file parameter, 317
  CLUSTER_DOMAIN
    ContinentalClusters configuration file parameter, 317
  CLUSTER_EVENT
    ContinentalClusters configuration file parameter, 317
  CLUSTER_NAME
    ContinentalClusters configuration file parameter, 318
  CLUSTER_TYPE
    ContinentalClusters configuration file parameter, 318
    in MetroCluster/CA, 306
    in MetroCluster/SRDF, 312
  cmapplycond
Index

ContinentalClusters command, 323
cmcheckcond
ContinentalClusters command, 323
cmdrmond
ContinentalClusters daemon, 324
cmdsentryd
ContinentalClusters daemon, 324
cmddeletecond
ContinentalClusters daemon, 324
cmddeletecond command, 201
cmomd
Object Manager daemon, 325
cmquerycond
ContinentalClusters command, 325
cmreadlog
ContinentalClusters command, 325
cmrecoverd, 186
ContinentalClusters command, 326
how the command works in ContinentalClusters, 188
command line
cmrecoverd, 186
symdg, 112
symgate, 114
command line interface, EMC Symmetrix, 105
concepts in ContinentalClusters, 132
configuring, 35
a three-data-center architecture, 55, 92
additional nodes in ContinentalClusters, 194
arbiter nodes, 56, 93
configuring, 153
ContinentalCluster Recovery cluster hardware, 156
ContinentalClusters recovery cluster, 156
data replication for ContinentalClusters, 153
disaster tolerant Ethernet networks, 35
disaster tolerant FDDI networks, 34
disaster tolerant WAN, 36
gatekeeper devices, 114
hardware for MetroCluster, 64, 101
monitoring in ContinentalClusters
monitor packages in ContinentalClusters, 162
Primary cluster, 154
verifying EMC Symmetrix configuration, 115
verifying XP series configuration, 76
configuring continental clusters, 163
configuring for ContinentalClusters, 153
configuring MetroCluster, 123
ContinentalClusters installing, 159
consistency of data, 26
CONSISTENCYGROUPS in MetroCluster/SRDF, 313
continental cluster, 22, 145, 153
building, 131
deleting, 201
renaming, 202
Continental Clusters overview, 153
CONTINENTAL_CLUSTER_NAME
ContinentalClusters configuration file parameter, 318
ContinentalClusters, 131
checking status, 197
concepts, 132
configuration file, 163
log files, 200
maintenance tasks, 194
monitor package, 162
monitoring, 163
switching clusters, 185
validating the configuration, 179
ContinentalClusters configuration checking and applying, 176
ContinentalClusters configuration file
cluster information, 164
monitoring definitions, 170
recovery groups, 166
Continuous Access XP in ContinentalClusters, 203
Continuous Access XP with ContinentalClusters
ContinentalCluster with Continuous Access XP, 153
creating
EMC Symmetrix device groups, 112
volume groups, 77, 115
currency of data, 26
D
DAS FDDI, configuring, 34
data center, 17
one, 53, 90
two, 53, 91
two-data-center architecture, 118
data consistency, 26
Index

data currency, 26
data recoverability, 26
data replication, 26, 153
  FibreChannel, 44
  ideal, 32
  logical, 30
  off-line, 26
  online, 27
  over WAN, 146
  physical, 27, 203, 233
restoring after a disaster, 190
synchronous or asynchronous, 27

data replication procedures for cascading failover, 281
DATA_RECEIVER_PACKAGE
ContinentalClusters configuration file parameter, 318
DATA_SENDER_PACKAGE
ContinentalClusters configuration file parameter, 319
DC1HOST
in MetroCluster/CA, 307
package control script variables, 81
DC2HOST
in MetroCluster/CA, 307
deleting a continental cluster, 201
Design of disaster tolerant cluster
MetroCluster/CA, 53
device groups creating, 112
device names
  EMC Symmetrix logical devices, 113
  mapping, 107
mapping Symmetrix to
  command line
  SymCLI database, 105
verify configuration, 115
disk command line interface, 105
device names, 106
serial number, 107
distributing MetroCluster configuration, 82, 126
Documenting the recovery procedure, 180
domino mode, 101
down state in ContinentalClusters, 136

E
EMC SRDF
cascading failover, 265
EMC SRDF in ContinentalClusters, 233
EMC SRDF with
  ContinentalClusters
  ContinentalCluster with EMC SRDF, 153
EMC Symmetrix
  creating device groups, 112
  device names, 106
gatekeeper devices, 114
R1 and R2, 91
serial number, 107
supported RAID configurations, 101
SymCLI database, 105
verify configuration, 115
EMC Symmetrix logical device names, 113
design state in ContinentalClusters, 136
Ethernet, disaster tolerant, 35
evaluating need for disaster tolerance, 14
events

Index
Index

in ContinentalClusters, 136
exporting volume groups, 77, 115

F
failback scenarios
in ContinentalClusters with Continuous Access XP, 224
in ContinentalClusters with EMC SRDF, 254
failover
cascading, 265
FDDI, disaster tolerant, 34
FENCE
in MetroCluster/CA, 307
FibreChannel clusters, 44
FibreChannel hub configuration for campus cluster, 44

G
gatekeeper devices, 114
geographic dispersion of nodes, 25

H
hardware configuration checklist for MetroCluster, 61, 97
hardware for ContinentalCluster Recovery cluster, 156
HORCMINST
in MetroCluster/CA, 307

I
importing volume groups, 77, 115
installing ContinentalClusters product, 159

L
local cluster, 18
log files reviewing in ContinentalClusters, 200
logical data replication, 30
logical device names, EMC Symmetrix, 113

M
M by N configurations setting up, 117
M by N configurations with MetroCluster/SRDF, 102
main control unit on XP series, 54
maintaining a continental cluster, 194
maintaining a MetroCluster/CA environment, 83
mapping EMC Symmetrix and HP-UX devices, 110, 113
mapping Symmetrix and HP-UX devices, 107
MAX_CONFIGURED_PACKAGES setting in MetroCluster/CA, 69
MC/ServiceGuard, 16
with ContinentalClusters, 131
with MetroCluster, 87
with MetroCluster/CA, 51
MC/ServiceGuard, package configuration, 79, 81, 124, 125
MetroCluster, 64, 69, 87, 100, 101
configuring, 123
MetroCluster with EMC SRDF, 90
MetroCluster/CA, 51
maintenance, 83
overview, 52, 88
metropolitan cluster, 53, 90
definition, 20
MirrorDisk/UX, 45
modifying
MetroCluster package control script, 79, 124
modifying ContinentalClusters configuration file, 163
MONITOR_INTERVAL
ContinentalClusters configuration file parameter, 319
MONITOR_PACKAGE_NAME
ContinentalClusters configuration file parameter, 319
monitoring, 178
receiving ContinentalClusters notification, 185
sample package configuration file for ContinentalClusters, 163
monitoring definitions entering in ContinentalClusters configuration file, 170
monitoring over a wide area network, 135
MONITORING_CLUSTER
ContinentalClusters configuration file parameter, 319
multiple points of failure, 17
MULTIPLE_PVOL_OR_SVOL_FRAMES_FOR_PKG
in MetroCluster/CA, 308
mutual recovery configuration, 133

N
network redundant, 53, 90
networks
disaster tolerant Ethernet, 35
disaster tolerant FDDI, 34
disaster tolerant WAN, 36
node
adding to ContinentalClusters, 194
NODE_NAME
ContinentalClusters configuration file parameter, 319
nodes
allowed in three-data-center architecture, 55, 92
arbitrator nodes, 56, 93
arbitrators, 91
used as arbitrators, 54
NOTIFICATION
ContinentalClusters configuration file parameter, 320
notifications entering in ContinentalClusters configuration file, 170
how they work, 138
receiving, 185
O
off-line data replication, 26
online data replication, 27
operations staff
for ContinentalClusters, 148
general guidelines, 40
overview of MetroCluster/CA, 52, 88
P
package
for ContinentalClusters monitoring, 163
worksheet, 63, 99
package configuration worksheet for MetroCluster/CA, 63
package control script
distributing MetroCluster script to nodes, 82, 126
package control script configuration, 79, 124
package PKGR1SYMGRP, 125
package R1 and R2
package control script variables, 125
package switching disabling in ContinentalClusters, 141
package switching via cmrecovery command, 186
packages configuring for disaster recovery with MetroCluster/CA, 79
differences between ServiceGuard and ContinentalClusters, 141
PATH in MetroCluster/SRDF, 313
physical data replication, 27
using Continuous Access XP in ContinentalClusters, 203
using EMC SRDF in ContinentalClusters, 233
PKG_SWITCHING_ENABLED (AUTO_RUN) setting to NO in a continental cluster, 155
PKGDIR
in MetroCluster/CA, 309
in MetroCluster/SRDF, 313
PKGR1SYMGRP
in MetroCluster/SRDF, 313
PKGR2SYMGRP
in MetroCluster/SRDF, 313
planning continental cluster, 148
point-to-point FibreChannel configuration for campus cluster, 44
post.cmapply script, 127
power planning worksheet, 150, 151, 152
power sources redundant, 32
pre.cmquery script, 127
Primary cluster configuring, 154
primary cluster, 132
PRIMARY PACKAGE ContinentalClusters configuration file parameter, 321
PVOL and SVOL, 64
Q
quorum, 57, 93
R
R1 and R2, 91, 101
package control script variables, 125
R1HOST
in MetroCluster/SRDF, 314
R2HOST
in MetroCluster/SRDF, 314
RAID
supported configurations, 101
Raid Manager
creating configuration files in MetroCluster/CA, 70
sample configuration file, 73
recoverability of data, 26
recovery cluster, 132, 156
recovery group adding, 195, 196
defining in ContinentalClusters, 166
recovery procedure
Index

documenting, 180

testing  
testing the recovery procedure, 181

RECOVERY_GROUP_NAME  
ContinentalClusters  
configuration file  
parameter, 321

RECOVERY_PACKAGE  
ContinentalClusters  
configuration file  
parameter, 321

redundant power sources, 32

remote control unit on XP series,  
54

remote protection  
restoring in MetroCluster/CA  
after failures, 84, 231

renaming a continental cluster,  
202

replicating data, 26  
off-line, 26  
online, 27

over WAN, 146

required hardware, 64, 101

required software, 69, 100

restoring remote protection after failures with  
MetroCluster/CA, 84, 231

restoring SRDF links, 127  
RETRY, 126

in MetroCluster/SRDF, 314  
RETRYTIME, 126

in MetroCluster/SRDF, 314

rolling disaster, 40

rolling disasters, 37

RUN_SCRIPT_TIMEOUT, 124

S  
sample scripts, 100

SAS FDDI, configuring, 34

scripts  
automating configuration, 100  
post.cmapply for MetroCluster,  
127  
pre.cmapply for MetroCluster,  
127

security files

in ContinentalClusters, 159  
serial number, EMC Symmetrix,  
107

ServiceGuard cluster  
preparing for  
MetroCluster/CA, 64

data center, 53, 90

single point of failure, 46, 90

site coordination

for ContinentalClusters  
notification, 185

special device files, mapping to  
EMC Symmetrix, 110

split brain syndrome, 49

splitting SRDF links, 127

splitting SRDF links during  
volume group configuration,  
116

SRDF links  
restoring, 127

splitting, 116

splitting for package  
configuration, 127

starting for  
ContinentalClusters, 178

starting the monitor packaing in  
ContinentalClusters, 178

status

checking status of  
ContinentalClusters  
objects, 197

supported configurations on XP  
series, 65

SureStore E Disk Array XP  
series

with MetroCluster, 51

SVOL and PVOL, 64

switching to a recovery cluster  
using ContinentalClusters,  
185

SymCLI database, 105

symdg, 112

symgate command, 114

symld command, 113

synchronous data replication, 27

T  
testing  
ContinentalClusters  
configuration, 179

three-data-center architecture,  
53, 90

U  
unreachable  
state in ContinentalClusters,  
136

up  
state in ContinentalClusters,  
136

V  
validating the  
ContinentalClusters  
configuration, 179

verifying the need for recovery in  
ContinentalClusters, 186

volume groups  
creating, 77, 115

importing and exporting, 77,  
115

W  
WAITTIME  
in MetroCluster/CA, 309

WAN configuration, 36

wide area cluster, 22, 132

wide area network
used for monitoring in
ContinentalClusters, 135
worksheet
cluster configuration with
MetroCluster/CA, 62
ContinentalClusters, 149
power supply configuration,
150, 151, 152
worksheet for package
configuration
MetroCluster/CA, 63
worksheet, MetroCluster, 62, 98
worksheet, package, 63, 99

X
XP series
supported configurations, 65
verify configuration, 76
XP series disk array
with MetroCluster, 51