

Content Manager Backup/Recovery and High Availability: Strategies, Options, and Procedures

Introducing basic concepts, strategies,
options, and procedures

Addressing business continuity
and disaster recovery issues

Providing practical case
studies



**Wei-Dong Jackie Zhu
Julian Cerruti
Antony A. Genta
Holger Koenig
Hernán Schiavi
Thomas Talone**



International Technical Support Organization

**Content Manager Backup/Recovery and High
Availability: Strategies, Options, and Procedures**

March 2004

Archived

Note: Before using this information and the product it supports, read the information in “Notices” on page vii.

First Edition (March 2004)

This edition applies to Version 8, Release 2, of IBM DB2 Content Manager for Multiplatforms (product number 5724-B19) and Version 8, Release 2, of IBM DB2 Information Integrator for Content for Multiplatforms (product number 5724-B43).

© Copyright International Business Machines Corporation 2004. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	vii
Trademarks	viii
Preface	ix
The team that wrote this redbook	ix
Become a published author	xi
Comments welcome	xi
Chapter 1. Introduction	1
1.1 Introducing Content Manager	2
1.1.1 Architecture	3
1.1.2 Library Server	3
1.1.3 Resource Managers	3
1.1.4 Mid-tier server	4
1.1.5 Clients	4
1.2 Fundamentals	4
Chapter 2. Backup and recovery strategies and options	7
2.1 Backup requirements	8
2.1.1 Types of events	8
2.1.2 Speed of recovery	10
2.1.3 Backup windows	10
2.1.4 Recovery points	10
2.1.5 Units of recovery	11
2.1.6 Backup of Content Manager supporting files	11
2.2 Backup options for Content Manager	11
2.2.1 Cold backup	13
2.2.2 Warm backup	13
2.2.3 Hot backup	14
2.3 Library Server and DB2 database backup and recovery	14
2.3.1 DB2 logging concepts	15
2.3.2 Backing up a DB2 database	21
2.3.3 Restoring a DB2 database	28
2.4 Resource Manager backup and recovery	29
2.4.1 Resource Manager data areas	30
2.4.2 Resource Manager services	32
2.4.3 Backup and recovery of the storage area	39
2.5 Content Manager configuration	45
2.6 Additional considerations for Content Manager	46

2.6.1	Stopping Content Manager activity	46
2.6.2	Library Server extensions	52
2.6.3	Resource Manager extensions	55
2.6.4	Validation utilities	55
2.6.5	Tivoli Storage Manager backup	56
2.6.6	Information Integrator for Content configuration database	58
2.6.7	Content Manager eClient	59
2.6.8	User applications and custom code	59
Chapter 3.	Practical backup and recovery procedures	61
3.1	Component overview	62
3.1.1	Library Server	62
3.1.2	Resource Manager	64
3.1.3	Summary of components	65
3.2	Planning for the backup	66
3.3	Offline backup	68
3.3.1	Preparing an offline backup	68
3.3.2	Offline backup to a file system	69
3.3.3	Offline backup to Tivoli Storage Manager	72
3.3.4	Finishing and restarting	80
3.3.5	Restoring	81
3.4	Online backup	82
3.4.1	Preparing	82
3.4.2	Backup procedure	83
3.4.3	Restore	84
3.5	Multiple node configurations	86
3.6	Tivoli Storage Manager backup procedure	87
Chapter 4.	High availability strategies and options	91
4.1	Overview	92
4.1.1	Availability concept	92
4.1.2	Planned versus unplanned outages	93
4.1.3	High availability and continuous availability	93
4.1.4	Cost versus loss	94
4.1.5	Availability matrix	95
4.1.6	Levels of availability	96
4.1.7	Measuring availability	99
4.2	Content Manager HA strategies and options	100
4.2.1	Content Manager infrastructure fundamentals	101
4.2.2	High availability example 1: Clustering	104
4.2.3	Example 1 modified: Clustering with mid-tier RM application	108
4.2.4	High availability example 2: Clustering and replication	110
4.2.5	High availability example 3: Replication	113

4.2.6 High availability example 4: Single server configuration	118
4.2.7 High availability summary chart	119
Chapter 5. Practical procedures for high availability	121
5.1 Introduction	122
5.1.1 Out test case scenario	123
5.2 Library Server HACMP	125
5.2.1 Software installation	126
5.2.2 Library Server parameter values	127
5.2.3 File system setup and user IDs creation	129
5.2.4 DB2 and Content Manager instance creation on primary node . . .	137
5.2.5 Setting up shared disks and LVM	142
5.2.6 DB2 and Content Manager instance creation on secondary node .	147
5.2.7 HACMP topology configuration	149
5.2.8 Defining resource groups	160
5.2.9 HACMP post-configuration procedures	166
5.3 Resource Manager replication	168
5.3.1 Initial parameter definitions	168
5.3.2 Resource Manager installation and configuration	169
5.3.3 Cross-referencing server definitions	171
5.3.4 Enable collections for replication	173
5.3.5 Adjust replication schedule	175
5.3.6 Adjust Resource Manager fail-over settings	177
5.3.7 Replication test	178
5.4 The rest of the environment	179
5.4.1 Tivoli Storage Manager	179
5.4.2 Content Manager clients	180
5.5 Fail-over tests and results	181
5.5.1 Library Server only failover	182
5.5.2 Resource Manager failover and fallback	186
5.5.3 Simultaneous Library Server and Resource Manager failover . . .	192
Chapter 6. Business continuity and disaster recovery strategies	195
6.1 Overview	196
6.1.1 Business continuity and disaster recovery concept	196
6.1.2 Major disastrous events	198
6.1.3 Data protection cost	199
6.1.4 Trends in business continuity planning	200
6.1.5 Critical considerations and success factors	201
6.2 Business continuity strategies and options	201
6.2.1 Insourced	202
6.2.2 Outsourced	202
6.2.3 Business continuity needs assessment	202

6.3 Disaster recovery best practices	203
6.3.1 Disaster recovery: Best practices with Content Manager	205
6.3.2 Content Manager recovery	208
Chapter 7. Case study: Retirement Application Processing System . . .	219
7.1 The business problem	220
7.2 Solution overview	220
7.3 Solution design	223
7.3.1 High availability	223
7.3.2 Backup and recovery	227
7.3.3 Geographic distribution	229
Chapter 8. Case study IBM ECM solution: Personnel Records System.	231
8.1 The business problem	232
8.2 Solution definition and benefits	232
8.3 Scope and complexity	232
8.4 Solution architecture	234
8.5 Application and data protection strategy	235
8.5.1 Disaster recovery strategy	236
8.5.2 Backup and restore	236
8.5.3 System availability strategy	236
8.5.4 Full recovery	242
8.5.5 Incremental recovery	242
Appendix A. Sample scripts and programs	245
A.1 HACMP Library Server startup and shutdown scripts	246
A.1.1 Library Server startup script	246
A.1.2 Library Server shutdown script	248
A.2 Sample custom API program	249
Related publications	253
IBM Redbooks	253
Other publications	253
Online resources	254
How to get IBM Redbooks	254
Help from IBM	254
Index	255

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.


This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AIX®	ibm.com®	TotalStorage®
AS/400®	Predictive Failure Analysis®	VideoCharger™
Chipkill™	pSeries®	VisualAge®
DB2 Universal Database™	Redbooks™	WebSphere®
DB2®	Redbooks (logo)  ™	xSeries®
Enterprise Storage Server®	RS/6000®	z/OS®
@server®	SP2®	
IBM®	Tivoli®	

The following terms are trademarks of other companies:

Intel, Intel Inside (logos), MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Preface

For a number of years, businesses and government agencies have implemented various technologies for managing and sharing business documents and unstructured data. Although they have served well, structured and unstructured data continues to increase, data retention requirements and user access requirements continue to change, and the demand for the readiness and availability of business systems and data becomes even higher. As most organizations have discovered, the use of content management systems is no longer an optional or discretionary item, it is vital and necessary; it is what makes an organization's success viable.

To a business that has become dependent on their content management systems, the availability of these systems is of crucial importance. Several technologies of various degrees have provided an answer to backup, availability, and disaster recovery requirements, but all at a price. How can you achieve maximum availability of your content systems while balancing costs, resources, and skills? The purpose of this IBM® Redbook is to introduce the concepts of backup/recovery, high availability, and disaster recovery for IBM DB2® Content Manager systems, and provide strategies, options, and implementation steps to protect your Content Manager systems. This redbook also will help IT architects, specialists, project managers, and decision makers identify the best high availability and disaster recovery strategies, and integrate them into the Content Manager solution design process.

The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization, San Jose Center.

Wei-Dong Jackie Zhu is a Content Manager Project Leader with the International Technical Support Organization at the Almaden Research Center in San Jose, California. She has more than 10 years of software development experience in accounting, image workflow processing, and digital media distribution. She holds a Master of Science degree in Computer Science from the University of the Southern California. Jackie joined IBM in 1996.

Julian Cerruti is an Advisory IT Specialist in IBM Argentina. He has five years of experience with IBM content management solutions and has been working at IBM for more than seven years. He is currently working as a technical specialist in IBM Software Group, responsible of supporting the sales of the whole IBM content management software portfolio for customers and Business Partners in Argentina, Paraguay and Uruguay. Previously, he has worked with IBM Global Services, leading and delivering the implementation of Content Manager, Content Manager OnDemand, and Tivoli® Storage Manager solutions. Julian holds a master's degree in Electrical Engineering from University of Buenos Aires. His main areas of interest include innovative information technology research and development.

Antony A. Genta is a Consulting IT Specialist with the IBM Federal Software Sales Team in Bethesda, MD, U.S. He has 19 years of experience in the data and content management field. He has worked at IBM for three years. His areas of expertise include e-business solution architecture, project planning and management, enterprise content management, data management and GIS solution design, and digital media. He has written extensively about enterprise content management solutions, DBMS and GIS systems, and strategic information systems. Before joining the IBM three years ago, Mr. Genta directed an Enterprise Content Management Practice at Battelle Memorial Institute in Washington, DC.

Holger Koenig is an IT Specialist with IBM Information Management development in Boeblingen, Germany. He has five years of experience in content management, working in pre-sales, services, and now in Business Partner enablement teams. He teaches technical classes and certification workshops for partners about IBM DB2 Content Manager products in EMEA. His area of expertise includes DB2 UDB and all Content Manager products on UNIX® and Microsoft® Windows® platforms.

Hernán Schiavi is an IT Specialist of Services and Support for pSeries® AIX® in SSA ITS SW at IBM Argentina. He has five years of experience in installation, configuration, implementation, administration, problems resolution, and support for AIX and SP2® systems. He also worked in the pre-sales area for the RS/6000® servers, focused mainly on architecture and configurations. Hernán's area of expertise includes implementations of high availability (HACMP) and performance analysis for large systems. He also provide support for IBM Tivoli software with the IBM Tivoli Storage Manager product.

Thomas Talone is a Certified Consulting Software Architect with 18 years of technical implementation, consultative, and customer strategic planning experience with IBM. He specializes in large scale enterprise content management solutions, infrastructure architectures, and high availability designs for IBM content management products. Currently, he is a Software Technical

Engineer for the IBM Content Management Software Group. He holds a bachelor's degree in Computer Science and an M.B.A. degree in Finance. He has written and presented on high availability and disaster recovery strategies for IBM DB2 Content Manager products at internal and customers conferences for the IBM Americas.

Thanks to the following people for their contributions to this project:

Emma Jacob
International Technical Support Organization, San Jose Center

Mario Lichtsinn
Cataldo Mega
Chunguang Zheng
Leonora Wang
IBM Software Group in the U.S.

Eva Billich
Holger Juschke
IBM Germany

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our Redbooks™ to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

- ▶ Use the online **Contact us** review redbook form found at:

ibm.com/redbooks

- ▶ Send your comments in an Internet note to:

redbook@us.ibm.com

- ▶ Mail your comments to:

IBM Corporation, International Technical Support Organization
Dept. QXXE Building 80-E2
650 Harry Road
San Jose, California 95120-6099

Introduction

In this chapter, we introduce the concepts, definitions and strategies discussed throughout this redbook for backup and recovery (BR), high availability (HA) and disaster recovery (DR) as it pertains to IBM DB2 Content Manager Version 8 in a multiplatform environment (Microsoft Windows, AIX, Sun Solaris, and Linux).

1.1 Introducing Content Manager

This section introduces the IBM DB2 Content Manager Version 8 product (Content Manager) and its components that are covered in this redbook as they relate to backup, high availability, and disaster recovery. It is important to first have an understanding of the Content Manager architecture and subsystems before diving into the topics of this book.

Content Manager provides a scalable, enterprise-wide repository system for the capture, creation, organization, management, workflow routing, archiving, and life cycle management of business content. It handles sharing, reuse, and archiving of all types of digitized content. The digitized content supported by Content Manager includes HTML and XML-based Web content, images, electronic office documents, and rich media, such as digital audio and video. Content Manager uses a triangular architecture, as shown in Figure 1-1.

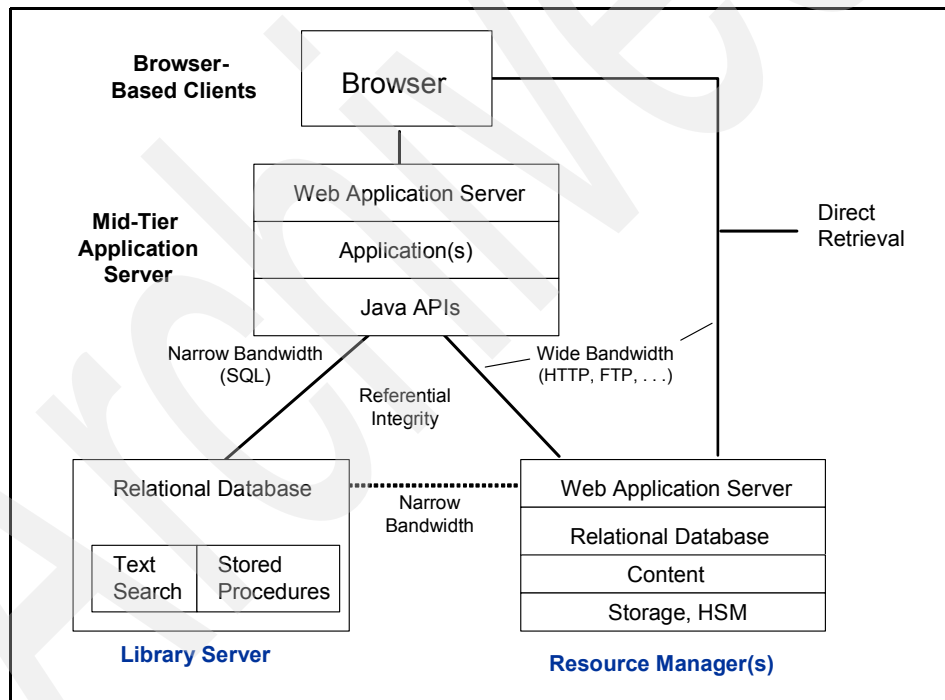


Figure 1-1 Content Manager Version 8 logical model

1.1.1 Architecture

Content Manager is built on multi-tier distributed architecture, with a Library Server that manages, indexes, and searches documents, Resource Managers that manage the actual digitized objects, a mid-tier server that acts as a broker between the client and the Library Server, and Windows-based and browser-based clients that provide the graphical end-user interface to the servers. Client applications use a single object-oriented application programming interface (API) to invoke all Content Manager services, which are divided between the Library Server and one or more Resource Managers. A single implementation supports a single Library Server, along with multiple Resource Managers and clients. Multiple, distinct applications of Content Manager can be installed on a single physical server.

Content Manager is available for IBM @server® iSeries (AS/400®) and z/OS® environments. Content Manager for Multiplatforms and Content Manager for z/OS Version 8.2 have the same capabilities, with minor differences. For example, Content Manager for z/OS currently lacks full-text search, which is planned for the next release.

1.1.2 Library Server

The Library Server manages the content metadata and is responsible for access control to all content. It maintains the indexing information for all multimedia content held in a Resource Manager. Users submit requests through the Library Server. The Library Server validates the access rights of the requesting client and then authorizes the client to directly access the object in the designated Resource Manager. The Library Server also maintains referential integrity between the indexing information and the objects themselves. The Library Server is built on IBM DB2 relational database management system (RDBMS) or Oracle. All access to the Library Server is via the database query language SQL, and all Library Server logic runs within DB2. With Content Manager, no persistent processes operate on the Library Server; all content management functions are stored procedures executed by DB2. Content metadata in the Library Server is backed up and recovered using standard database tools.

1.1.3 Resource Managers

Resource Managers are the repositories that contain the digitized content and manage the storage and retrieval of objects. The Resource Manager supports caching, replication and provides hierarchical storage management when used in conjunction with IBM Tivoli Storage Manager. A single Resource Manager can manage multiple VideoCharger™ systems as well. The Resource Manager architecture provides an extensible model that enables the support of additional Resource Managers in the future.

1.1.4 Mid-tier server

The mid-tier server functions as a broker that mediates communications between the client and the Library Server. It manages connections to the Library Server and, optionally, to the Resource Managers.

1.1.5 Clients

Users can access Content Manager repositories through Windows clients (thick client) or an eClient (thin client). The eClient Web application consists of JavaServer Pages (JSP), servlets, and a viewer applet that runs on IBM WebSphere® Application Server. The eClient can communicate directly with the Resource Manager using Internet protocols. It can talk directly to the application server (for example, WebSphere). The eClient provides federated access to and searches across all Content Manager and non-IBM repositories.

1.2 Fundamentals

In this redbook, we described three elements to protect and make available the services and data within a Content Manager environment:

- ▶ Backup and recovery (BR)
- ▶ High availability (HA)
- ▶ Disaster recovery (DR)

Figure 1-2 on page 5 provides the fundamental overview.

Fundamentals

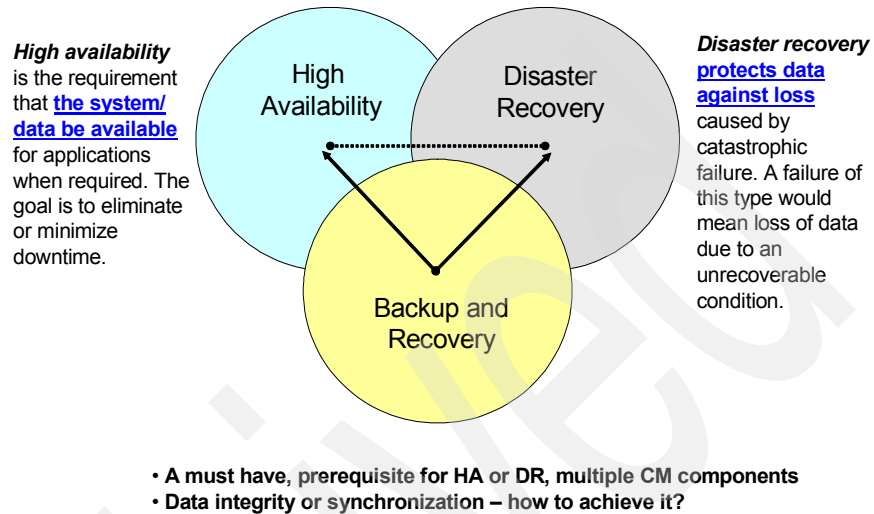


Figure 1-2 Fundamentals

Backup and recovery of a Content Manager system is a requirement to achieve high availability or disaster recovery, or both. It is also a stand-alone service that needs to be performed even if an implementation does not have the need for high availability or disaster recovery. Backup and recovery is a *must have* to protect the content itself, the database metadata, and the control data. Chapter 2, “Backup and recovery strategies and options” on page 7 discusses the different backup and recovery options within a Content Manager environment. Chapter 3, “Practical backup and recovery procedures” on page 61 describes the implementation and configuration steps for backup and recovery based on typical Content Manager configurations.

High availability, an optional service, is the requirement that the *system and data be available* during production hours, minimizing any downtime due to an unplanned outage. High availability is of vital importance as content-enabled applications evolve from the back office environment to client-facing, mission-critical applications. Chapter 4, “High availability strategies and options” on page 91 discusses the different high availability strategies that can be used to improve the system uptime due to an unplanned outage within a Content Manager environment, followed by implementation and configuration steps for

one example of a Content Manager high availability installation, detailed in Chapter 5, “Practical procedures for high availability” on page 121.

Disaster recovery, also an optional service, is aimed at *protecting against data loss* caused by a catastrophic system failure or a natural disaster. In Chapter 6, “Business continuity and disaster recovery strategies” on page 195, we discuss the best practices for disaster recovery strategies and options available for enterprise content management systems.

A Content Manager implementation can have a high availability strategy, a disaster recovery strategy, or both. Note, high availability focuses on system uptime, while disaster recovery focuses on data protection.

The remaining chapters of this book provide real-world case examples for high availability and disaster recovery implementations performed to improve the operating capabilities of a Content Manager environment.

Backup and recovery strategies and options

In this chapter, we describe different options about how to create backups of the Content Manager components. We guide you to the best backup strategy fitting your functional requirements, and we show you how to restore your data without losing data or leaving Content Manager components in an inconsistent state.

We show you planning steps for backup and recovery of:

- ▶ The Library Server and Resource Manager databases
- ▶ Files stored on the Resource Manager
- ▶ Content Manager configuration

In Chapter 3, “Practical backup and recovery procedures” on page 61, we describe the actual implementation steps based on typical Content Manager configurations.

2.1 Backup requirements

A backup strategy is one part of an overall data management plan. It is necessary to understand how important the data is to the function of the organization. Before planning for a backup strategy, there are other considerations that reduce the risk of a data loss:

- ▶ Redundant array of inexpensive disk (RAID) devices
- ▶ Dual access paths
- ▶ Dual I/O controllers
- ▶ Dual power supplies
- ▶ Backup of standby processors
- ▶ Uninterruptable power supplies

None of these on their own can guarantee the availability of data, but in combination, they can reduce the impact of a failure.

Before designing a backup strategy, the requirements that the strategy must satisfy need to be defined. Factors that need to be considered when defining the requirements for a backup strategy include:

- ▶ Types of events (the categories of incidents that might occur)
- ▶ Speed of recovery (how quickly you need to be able to recover)
- ▶ Backup windows (the periods of time at which backups can be formed)
- ▶ Recovery points (to which points in time you need to be able to recover)
- ▶ Units of recovery (which other tables and files need to be recovered to the same point in time)

2.1.1 Types of events

This section describes different categories of incidents that might occur. These include user error, statement failure, transaction failure, media failure, and disaster. We explain how to react to these kind of incidents and how to plan for them.

User error

The first type of event is the user error. For example, the user is new to the application and deletes some data. One way to restrict this is setting up a tight access control, but still there might be delete operations in error by a person having the authority to do so.

Content Manager and DB2 provide facilities that reduce the risk or impact of user errors:

- ▶ Use Content Manager and DB2 security to restrict access to the data.

- ▶ Restore the entire database to the point in time before the user error (updates might be lost).
- ▶ Restore the Resource Manager's storage area in case the data has been deleted there as well.

Statement failure

Content Manager operations and related SQL statements that are syntactically correct might fail, because, for example, the database is full. Content Manager will usually:

- ▶ Detect such problems.
- ▶ Roll back the effects of the failing statement.
- ▶ Report the problem to the user.

After the fundamental cause of the problem has been resolved, the user can retry the statement and continue to work. There is normally no need to take any special action to recover from statement failures.

Transaction failure

Transactions may fail for a variety of reasons:

- ▶ Programming errors
- ▶ Network failures
- ▶ Failures of the operating system or RDBMS
- ▶ Power failures

The actions required to recover from these situations vary according to the particular circumstances. However, Content Manager and DB2 will ensure that the integrity of the data it manages is preserved. There is no need to restore data to recover from transaction failures.

Media failure

Content Manager and DB2 normally use magnetic disk as the medium on which they store the data that they manage. If a disk volume is physically damaged or destroyed, at a minimum, it is necessary to restore the data files that have been lost to the state they were in when they were last backed up.

Disaster

Many organizations have developed plans for recovery from disasters such as:

- ▶ Floods
- ▶ Fires
- ▶ Accidents
- ▶ Earthquakes
- ▶ Terrorist attacks

You need to ensure that your strategy for backing up and recovering data fits in with any such plans. For example, arrangements to create backups to a removable medium or stored off-site should be made. The subject of disaster recovery is very broad and is discussed in more detail in Chapter 6, “Business continuity and disaster recovery strategies” on page 195.

2.1.2 Speed of recovery

The actual time taken for recovery depends on a number of factors, some of which are outside of the administrators control (for example, hardware might need to be repaired or replaced). Nevertheless, there are certain things that can be controlled and that will help to ensure that recovery time is acceptable:

- ▶ Develop a strategy that strikes the right balance between the cost of backup and the speed of recovery.
- ▶ Document the procedures necessary to recover from the loss of different groups or types of data files.
- ▶ Estimate the time required to execute these procedures. Do not forget the time involved in identifying the problem and the solution.

Set user expectations realistically, for example, by publishing service levels that you are confident you can achieve.

2.1.3 Backup windows

Some environments do not allow databases and content to be backed up while they are in use. In such cases, the components have to be shut down before the backup starts, and cannot be restarted until after the backup has completed. Shutting down Content Manager often means that users cannot use applications. It is important to ensure that the times at which Content Manager is shut down and unavailable are acceptable to the users.

Even if it is possible to perform backups while the system is operational, you need to ensure that any load on processors or networks caused by the backup process does not result in performance or response degradation that is unacceptable to the end users.

2.1.4 Recovery points

You need to define the points in time to which you will restore data. For example, you might need to recover the data:

- ▶ To the state it was in when the last transaction was completed
- ▶ To a consistent state that is no more than 24 hours old

In addition to either of these, there might be, for example, the requirement:

- ▶ To restore individual data to the state it was in at any particular date within the last 30 days

Whatever your situation, you need to consider recovery points and define a policy that is both achievable and acceptable to your user community.

2.1.5 Units of recovery

In most circumstances, it might not be sufficient to restore databases or storage areas to the state they were in at some point in the past. Often, in order to *maintain data consistency*, there is the need to restore data held in databases or files that have not been lost or damaged. This *undamaged* data needs to be restored to the same point in time as the *damaged* data. In developing a backup strategy, understanding the relationships between the data objects on which user applications rely is important. The key point is that the backup and recovery strategy must take into account the needs of the applications that use the data.

2.1.6 Backup of Content Manager supporting files

Content Manager consists of many different parts, because there are at least one Library Server and one Resource Manager database. In addition, you will find closely related data, such as the access modules for the Library Server, full text indexes, or the storage areas on the Resource Manager. These files are required for the operation of Content Manager and need to be backed up!

Additional files are:

- ▶ Initialization parameter files
- ▶ Password file
- ▶ Files that define the environment
- ▶ Network configuration files

They are external files and are not part of the database, because they must be accessible for reading, or even editing, when the database is down. The backup strategy must ensure that these files are also backed up using operating system or third-party tools such as Tivoli Storage Manager.

2.2 Backup options for Content Manager

With the different requirements, as shown in 2.1, “Backup requirements” on page 8, there will be many different implementations for backup and recovery options. Figure 2-1 on page 12 shows different tiers, starting with very simple

requirements for tier 1 and getting more and more complex to tier 5. The higher the requirements are, the higher will be the need of time, effort, and funding.

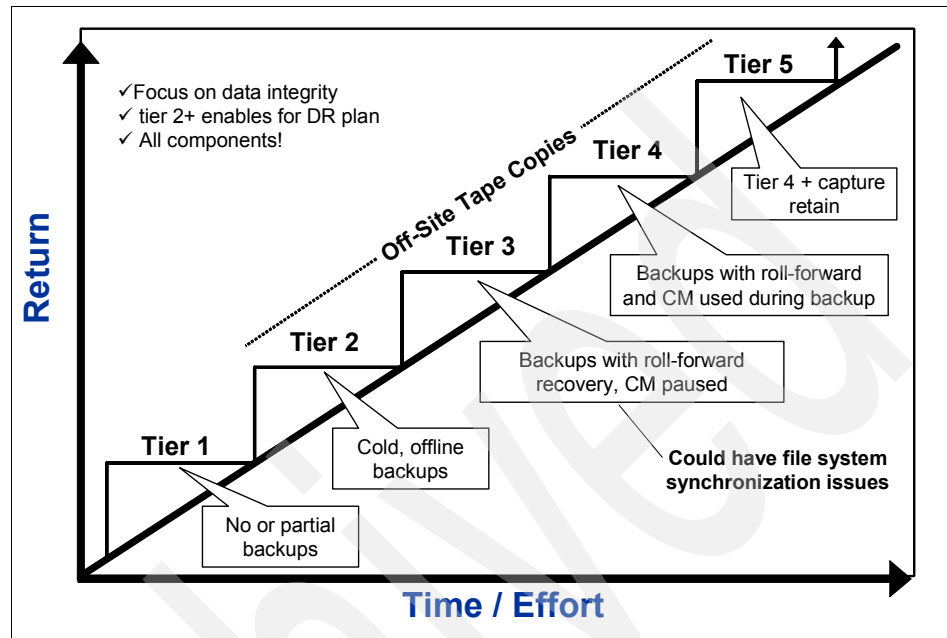


Figure 2-1 Backup tiers for Content Manager

Tier 1 implements no or only a very basic backup strategy. It might consist of backing up part of a file system using for example standard operating system tools.

Tier 2 describes the simplest complete backup option for Content Manager. First, all Content Manager components are stopped, therefore ensuring that no activity happens on any of the Content Manager system components. The database is backed up using DB2 tools, and all other components are backed up as files using operating system tools, Tivoli Storage Manager, or other backup tools of your choice. With tier 2, it is possible to restore to exactly the point in time of the backup.

In extension to tier 2, tier 3 also stores database recovery log files. With these files, it is possible to roll forward the databases to a point in time later than the last offline backup. With this, tier 3 reduces the amount of data that might be lost, but also adds complexity to synchronize the databases with the file systems.

Tier 4 extends the backup options to a more highly available system. Using tier 4, it is not necessary to stop user activity on the system anymore. Tier 4 requires access to archived database log files.

To achieve the highest level, it is necessary to capture all imported data after the last backup. Most importing subcomponents, such as Kofax Ascent Capture, have the ability to save their release scripts and data, so it is possible to reload any data added after the last backup. Tier 5 also needs attention on the users' or application builders' side.

There are several backup options:

- ▶ Cold backup
- ▶ Warm backup
- ▶ Hot backup

We discuss each in the following sections.

2.2.1 Cold backup

To create a *cold backup* of your system, it is important to stop all activity on your Content Manager system. This requires that all users are disconnected from the Library Server and mid-tier server and that all components for the Resource Manager are being shut down. This includes the Resource Manager Web application, as well as the Resource Manager services such as migrator, stager, purger, and replicator and also the Library Server monitor. No connection to any Content Manager database is allowed nor should any files be accessed by an application.

After all components are stopped, the backup needs to be performed. It is necessary to back up the Content Manager server databases, Library Server access modules, Net Search Extenders (NSE) text indexes, Resource Manager storage area and important configuration files.

Having completed all the backup operations, all the Content Manager components need to be restarted and users can log on to the system again.

With a cold backup, it is easy to manage distributed installations, such as having the Library Server and Resource Managers on different physical machines. The requirement is that all of the components are stopped and backed up at the same time. A cold backup corresponds to tier 2 in Figure 2-1 on page 12.

2.2.2 Warm backup

A *warm backup* typically means that users are active on the system while the backup operation is being performed. However, to create a consistent, point-in-time backup of all the components, while users can do all kinds of activities, is very complex.

We want to define a warm backup for Content Manager where no activity is performed on any Content Manager data file besides the databases. This can be accomplished by stopping all import, delete, and update operations from the users' side, while still maintaining read access. And of course, it is necessary to stop server-side activities such as migration, replication, or full text indexing.

With this, it is possible to create a point-in-time snapshot of all data files, and with the database in archival logging mode, it is possible to recover the Library Server and Resource Manager databases to exactly this point in time.

The warm backup option corresponds to tier 3 in Figure 2-1 on page 12. Creating a consistent backup of distributed Content Manager systems becomes more complicated, because the time stamps need to be exactly synchronized.

2.2.3 Hot backup

A *hot backup* corresponds to tier 4 and tier 5 in Figure 2-1 on page 12. When using hot backups, no user activity is forbidden. All Content Manager services are completely available to the user. Internal services, such as migration or full text indexing, might be paused during the backup operation to make it easier to recover to a consistent state.

When using a hot backup, it is always necessary to perform additional verification steps for data integrity. Section 2.6.4, "Validation utilities" on page 55 describes the utilities provided.

2.3 Library Server and DB2 database backup and recovery

This section shows the different DB2 logging concepts, how they work, and how they are used to meet our requirements for backup and recovery of Content Manager. It also shows different backup options provided by DB2 and how to restore a backup image.

Whenever a DB2 database is being created, DB2 will create tablespaces and log files. The tablespaces will later store the data while the log files are being used to keep track of transactions. With log files, a database can, for example, support the rolling back of transactions, recovery after a crash, or a rollforward after a database restore.

After the installation of Content Manager, both the tablespaces and log files will be stored on the same disk. This is neither a good choice for performance, because tablespaces and log files updates will cause a lot of I/O operations at the same time, nor for a media failure. It is highly recommended that you move

the log files to a different physical disk, because this will help increase the performance of your system and ensures that you can recover all the data in case the disk containing the tablespaces or the disk containing the log files fails.

Important: Always separate database log files from the database itself! This decreases the chance of data loss and helps improve system performance.

Content Manager creates by default two system-managed tablespaces for the Library Server database. One stores all the definition data, such as item types, user IDs, and process that are defined, while the other one stores the actual metadata of your documents.

2.3.1 DB2 logging concepts

All DB2 databases have associated log files. DB2 knows two different kinds of logging: circular logging, which is the default for DB2 databases and also for Content Manager, and archival logging.

Circular logging

Circular logging supports nonrecoverable databases. It uses primary and secondary log files, as shown in Figure 2-2. During typical operation of the system, primary log files are being used. Every transaction is written to a log file. After all primary log files have been used in a round-robin fashion, DB2 will reuse the log files if all transactions in this log file are either committed or rolled back. In case there are no available primary log files, DB2 will temporary create secondary log files. There is also a limit on the number of secondary log files. If this limit is reached, for example because of a very long transaction, and no log file became available, a log full condition will occur, and DB2 will roll back the entire unit of work.

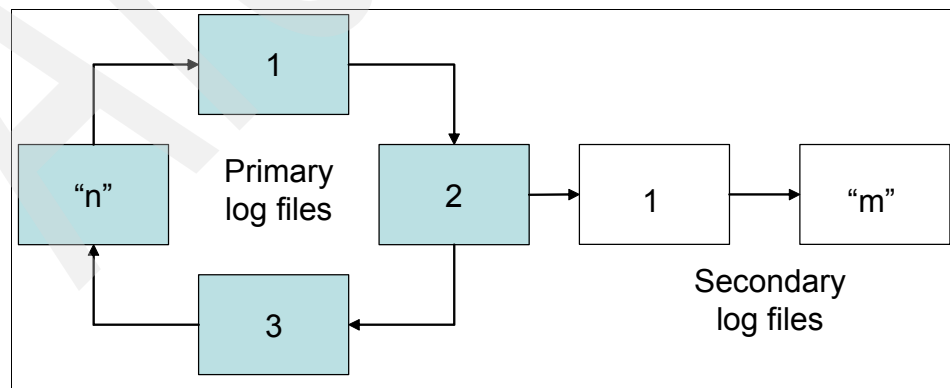


Figure 2-2 Circular logging

Primary log files are pre-allocated with the first connection to the database. In an opposite manner, secondary log files are created as needed.

Important: With circular logging, log files are overwritten after a certain amount of time. Because of that, you cannot roll forward a database after a restore operation when using circular logging. This also limits the backup capabilities to an offline backup.

To update the number of primary log files run:

```
db2 update database configuration for <database name> using LOGPRIMARY <num of primary log files>
```

To update the number of secondary log files run:

```
db2 update database configuration for <database name> using LOGSECOND <num of secondary log files>
```

To change the physical location of the log files, run:

```
db2 update database configuration for <database name> using NEWLOGPATH "<new log path>"
```

To query your current settings, run:

```
db2 get database configuration for <database name>
```

Look for the values in rows with LOGPRIMARY, LOGSECOND, and Path to log files. To check if circular logging is used, verify that LOGRETAIN and USEREXIT are set to OFF. A sample output is shown in Example 2-1. The important information for logging and backup and recovery in general are highlighted in bold.

Example 2-1 A sample database configuration

Database Configuration for Database ICMNLSDB	
Database configuration release level	= 0x0a00
Database release level	= 0x0a00
Database territory	= US
Database code page	= 1252
Database code set	= IBM-1252
Database country/region code	= 1
Dynamic SQL Query management	(DYN_QUERY_MGMT) = DISABLE
Discovery support for this database	(DISCOVER_DB) = ENABLE
Default query optimization class	(DFT_QUERYOPT) = 2

Degree of parallelism	(DFT_DEGREE) = 1
Continue upon arithmetic exceptions	(DFT_SQLMATHWARN) = NO
Default refresh age	(DFT_REFRESH_AGE) = 0
Number of frequent values retained	(NUM_FREQVALUES) = 10
Number of quantiles retained	(NUM_QUANTILES) = 20
Backup pending	= NO
Database is consistent	= NO
Rollforward pending	= NO
Restore pending	= NO
Multi-page file allocation enabled	= NO
Log retain for recovery status	= NO
User exit for logging status	= NO
Data Links Token Expiry Interval (sec)	(DL_EXPINT) = 60
Data Links Write Token Init Expiry Intvl	(DL_WT_IEXPINT) = 60
Data Links Number of Copies	(DL_NUM_COPIES) = 1
Data Links Time after Drop (days)	(DL_TIME_DROP) = 1
Data Links Token in Uppercase	(DL_UPPER) = NO
Data Links Token Algorithm	(DL_TOKEN) = MACO
Database heap (4KB)	(DBHEAP) = 2400
Size of database shared memory (4KB)	(DATABASE_MEMORY) = AUTOMATIC
Catalog cache size (4KB)	(CATALOGCACHE_SZ) = (MAXAPPLS*4)
Log buffer size (4KB)	(LOGBUFSZ) = 32
Utilities heap size (4KB)	(UTIL_HEAP_SZ) = 5000
Buffer pool size (pages)	(BUFFPAGE) = 250
Extended storage segments size (4KB)	(ESTORE_SEG_SZ) = 16000
Number of extended storage segments	(NUM_ESTORE_SEGS) = 0
Max storage for lock list (4KB)	(LOCKLIST) = 1000
Max size of appl. group mem set (4KB)	(APPGROUP_MEM_SZ) = 30000
Percent of mem for appl. group heap	(GROUPHEAP_RATIO) = 70
Max appl. control heap size (4KB)	(APP_CTL_HEAP_SZ) = 1000
Sort heap thres for shared sorts (4KB)	(SHEAPTHRES_SHR) = (SHEAPTHRES)
Sort list heap (4KB)	(SORTHEAP) = 256
SQL statement heap (4KB)	(STMTHEAP) = 16384
Default application heap (4KB)	(APPLHEAPSZ) = 1024
Package cache size (4KB)	(PCKCACHESZ) = (MAXAPPLS*8)
Statistics heap size (4KB)	(STAT_HEAP_SZ) = 4384
Interval for checking deadlock (ms)	(DLCHKTIME) = 10000
Percent. of lock lists per application	(MAXLOCKS) = 22
Lock timeout (sec)	(LOCKTIMEOUT) = 30

Changed pages threshold	(CHNGPGS_THRESH) = 60
Number of asynchronous page cleaners	(NUM_IOCLEANERS) = 1
Number of I/O servers	(NUM_IOSERVERS) = 3
Index sort flag	(INDEXSORT) = YES
Sequential detect flag	(SEQDETECT) = YES
Default prefetch size (pages)	(DFT_PREFETCH_SZ) = 16
Track modified pages	(TRACKMOD) = OFF
Default number of containers	= 1
Default tablespace extentsize (pages)	(DFT_EXTENT_SZ) = 32
Max number of active applications	(MAXAPPLS) = 200
Average number of active applications	(AVG_APPLS) = 5
Max DB files open per application	(MAXFILOP) = 64
Log file size (4KB)	(LOGFILSIZ) = 1000
Number of primary log files	(LOGPRIMARY) = 10
Number of secondary log files	(LOGSECOND) = 20
Changed path to log files	(NEWLOGPATH) =
Path to log files	= C:\DB2\NODE0000\
SQL00001\SQLLOGDIR\	
Overflow log path	(OVERFLOWLOGPATH) =
Mirror log path	(MIRRORLOGPATH) =
First active log file	=
Block log on disk full	(BLK_LOG_DSK_FUL) = NO
Percent of max active log space by transaction	(MAX_LOG) = 0
Num. of active log files for 1 active UOW	(NUM_LOG_SPAN) = 0
Group commit count	(MINCOMMIT) = 1
Percent log file reclaimed before soft ckcpt	(SOFTMAX) = 100
Log retain for recovery enabled	(LOGRETAIN) = OFF
User exit for logging enabled	(USEREXIT) = OFF
Auto restart enabled	(AUTORESTART) = ON
Index re-creation time	(INDEXREC) = SYSTEM (ACCESS)
Default number of loadrec sessions	(DFT_LOADREC_SES) = 1
Number of database backups to retain	(NUM_DB_BACKUPS) = 12
Recovery history retention (days)	(REC_HIS_RETENTN) = 366
TSM management class	(TSM_MGMTCLASS) =
TSM node name	(TSM_NODENAME) =
TSM owner	(TSM_OWNER) =
TSM password	(TSM_PASSWORD) =

Archival logging

Opposite to circular logging, archival logging does not overwrite existing log files. Instead, it creates new log files as needed. The three different log file conditions are shown and explained in Figure 2-3.

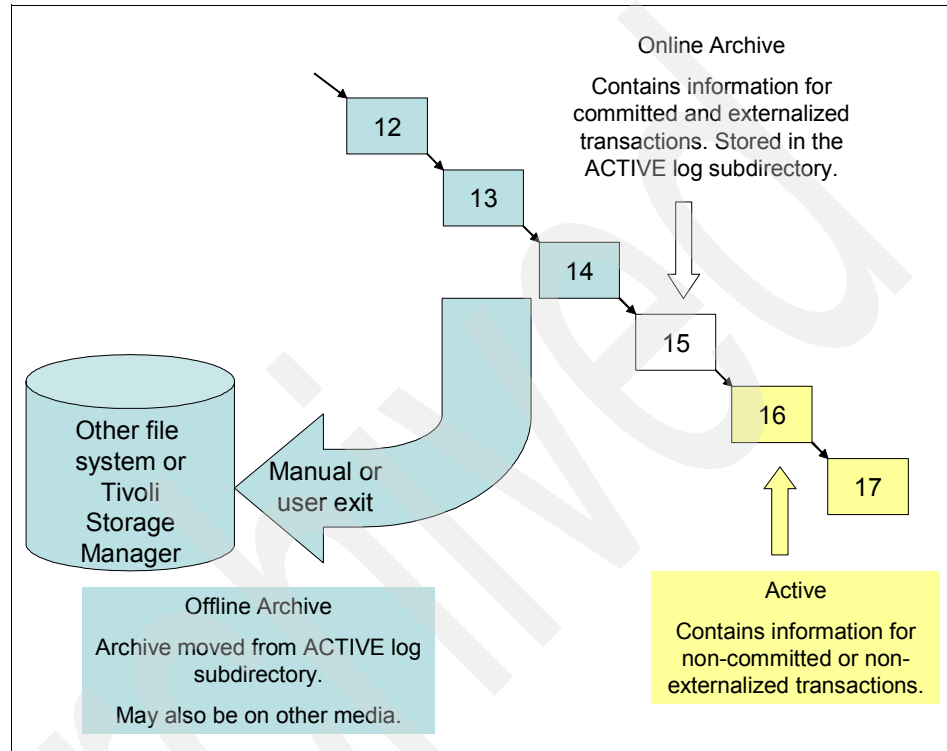


Figure 2-3 Archival logging

When using archival logging, you have the ability to repeat transactions to a database after a restore operation. This widely enhances the restore capabilities of DB2 databases and ensures no or only very little data loss when you need to restore a database.

On the other side, because over time more and more log files are created, your storage requirements increase, too. To move the log files not containing any non-committed or non-externalized transactions to a different storage location, DB2 offers a user exit, which is called by DB2 if a log file reaches this state. There are samples available for moving the log files to a different directory or to archive it to Tivoli Storage Manager, as shown in “Using user exit” on page 20.

To enable archival logging, run:

```
db2 update database configuration for <database name> using LOGRETAIN ON
```

Depending if there are currently no connections to this database, this setting becomes valid immediately, or if there are currently connections, it is necessary that all applications disconnect first. After completion, the database is going into backup pending status. It is mandatory to do a database backup at this time. A typical sequence of steps including a backup to a local file system is shown in Example 2-2.

Example 2-2 Enabling archival logging (LOGRETAIN)

```
db2 => update database configuration for ICMNLSDB using LOGRETAIN ON
DB20000I The UPDATE DATABASE CONFIGURATION command completed successfully.
SQL1363W One or more of the parameters submitted for immediate modification
were not changed dynamically. For these configuration parameters, all
applications must disconnect from this database before the changes become
effective.
```

```
db2 => force application all
DB20000I The FORCE APPLICATION command completed successfully.
DB21024I This command is asynchronous and may not be effective immediately.
```

```
db2 => connect to ICMNLSDB
SQL1116N A connection to or activation of database "ICMNLSDB" cannot be made
because of BACKUP PENDING. SQLSTATE=57019
```

```
db2 => backup database ICMNLSDB to c:\mybackups\
Backup successful. The timestamp for this backup image is : 20031017152657
```

```
db2 => connect to ICMNLSDB
```

Database Connection Information

Database server	= DB2/NT 8.1.3
SQL authorization ID	= ICMADMIN
Local database alias	= ICMNLSDB

Notice the backup pending status after enabling LOGRETAIN when trying to connect to the database. In this example, a full database backup to the directory c:\mybackups\ is being created.

Using user exit

To move the log files not containing any non-committed or non-externalized transactions to a different storage location, DB2 offers a user exit, which is called by DB2 if a log file reaches this state. There are samples available for moving the log files to a different directory or to archive it to Tivoli Storage Manager. On

Windows, the sequence of steps to prepare the user exit with Tivoli Storage Manager are:

1. Copy db2uext2.ctsm from C:\Program Files\IBM\SQLLIB\samples\c to a working directory and rename the file to db2uext2.c.
2. Verify and modify the Tivoli Storage Manager config values to be used (DSMI_DIR, DSMI_CONFIG, DSMI_LOG and MGT_CLASS).
3. Compile the user exit using:

```
c1 db2uext2.c -I c:\Progra~1\tivoli\tsm\api\include -link  
c:\Progra~1\tivoli\tsm\api\lib\tsmapi.lib
```

You need to have a C++ compiler, such as Microsoft Visual C++, installed. Note that you should not use any spaces in program directories in the command shown.
4. Copy the created .exe file to the SQLLIB\bin directory. We recommend that you back up the compiled user exit.

To enable user exit, run:

```
db2 update database configuration for <database name> using USEREXIT ON
```

When you use this user exit to move archived log files to Tivoli Storage Manager, the user exit will use a Tivoli Storage Manager archive copy group to decide where and how long to store each log file. With this in mind, you must configure the Tivoli Storage Manager server to include an archive copy group into the management class you will use to hold DB2 backup images and log files. In this archive copy group, you must specify the amount of time you want to keep DB2 archived log files before they are discarded.

Keep in mind that this setting must be coherent with the policies used for database backup images deletion (see “Maintenance of database backup copies” on page 25) and Resource Manager storage area retention periods (see “Server policies for storage area backup” on page 42).

2.3.2 Backing up a DB2 database

This section shows you how to create a backup of a DB2 database. Table 2-1 on page 22 shows you three different options for a full database backup and basic considerations for a restore operation.

Table 2-1 Full backup options

	Offline backup	Offline backup	Online backup
Logging type	Circular logging	Archival logging	Archival logging
Access during backup	N/A	N/A	Full (read and write)
Database state after restore	Consistent	Rollforward pending	Rollforward pending
Rollforward required	N/A	Any point in time after backup	Any point in time past backup

Full database backup: Offline

The easiest way to create a database backup is using an offline backup. This means that no application is accessing or connected to this database at this time. You can find out if any application is connected to your database by running:

```
db2 list application for database <database name>
```

If any application is still connected to the database, the response is:

Auth Id	Application Name	Appl. Handle	Application Id	DB Name	# of Agents
ICMADMIN	icmsvmig.exe	11	*LOCAL.DB2.00AA03122217	ICMNLSDDB	1
ICMADMIN	icmsvmig.exe	10	*LOCAL.DB2.00AA03122216	ICMNLSDDB	1
ICMADMIN	icmsvmig.exe	9	*LOCAL.DB2.00AA03122215	ICMNLSDDB	1
ICMADMIN	icmplsap.exe	2	*LOCAL.DB2.0053C3122156	ICMNLSDDB	1

We highly recommend that you end the connection by stopping the application. If it is necessary to end the application using DB2, use the command:

```
db2 force application <application handle>
```

Or to disconnect all applications, use:

```
db2 force application all
```

Note: The **force application all** commands terminate any connections to any database in the DB2 instance. If there is more than one database defined in an instance, use care not to interrupt other applications.

The message if no application is connected is:

```
SQL1611W No data was returned by Database System Monitor. SQLSTATE=00000
```

Offline backup is available for both circular and archival logging. To create an offline backup of a DB2 database, run:

```
db2 backup database <database name> to <directory>
```

Important: To create an offline backup, DB2 must not be stopped; all applications need to be disconnected from the database.

The output of this backup operation looks like this:

Backup successful. The timestamp for this backup image is : 20031015101505

Notice the time stamp shown at the end of the backup. You can use this time stamp to determine which database backup should be restored if a single directory contains more than one backup of the same database with different time stamps. Note, that the time stamp uses local time.

On Windows platforms, the backup is stored to files in a directory having naming conventions as shown in Figure 2-4.

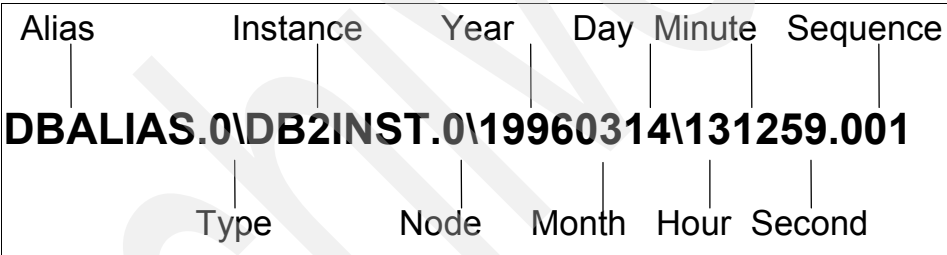


Figure 2-4 Backup naming conventions on Windows

On UNIX platforms, the backup is stored in a directory having naming conventions as shown in Figure 2-5.

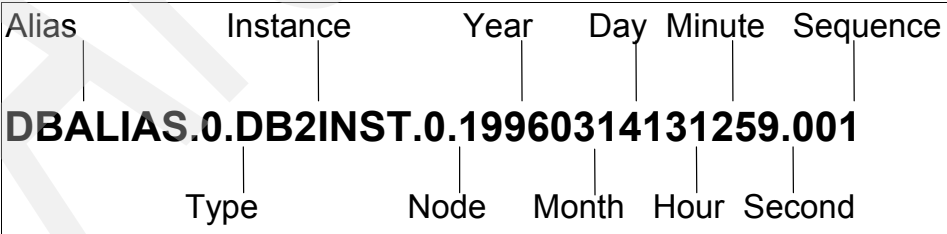


Figure 2-5 Backup naming conventions on UNIX

The main difference between the Windows and UNIX platforms is that on Windows, several subdirectories will be created, while on UNIX, just one folder name contains all the information.

Important: DB2 backups are platform specific. It is not possible to restore a backup to a different platform than the origin one.

Full database backup: Online

The advantage of using an online backup is that no application needs to disconnect from the database and no transaction needs to be stopped. The database remains accessible and able to be updated all time. For Content Manager, this means that there is no need to stop the Library Server nor the Resource Manager. Online backups are typically performed when systems need to be highly available and no downtime can be accepted.

The disadvantage while doing online backups is that database log files become very important. Because transactions will be in progress during the backup, the backup might contain these transactions in an uncommitted status. To be able to move the backup image to a consistent status, without open transactions, the log files are required.

Important: DB2 does support online backups for databases only if archival logging (LOGRETAIN) or USEREXIT has been enabled.

While using circular logging, DB2 cannot perform an online backup and returns the message: "SQL2413N Online backup is not allowed because either logretain or userexit for roll-forward is not activated, or a backup pending condition is in effect for the database."

To do an online backup, run:

```
db2 backup database <database name> ONLINE to <directory>
```

The output of this backup operation looks like this:

```
Backup successful. The timestamp for this backup image is : 20031015101505
```

Notice the time stamp shown at the end of the backup. You can use this time stamp to determine which database backup should be restored if a single directory contains more than one backup of the same database with different time stamps. Note that the time stamp uses local time.

Online backups are stored in a similar directory structure as offline backups, shown in "Full database backup: Offline" on page 22. Online backups are also platform specific and cannot be restored on a different platform.

Backup using Tivoli Storage Manager

DB2 has built-in capabilities to back up a database to Tivoli Storage Manager. By using Tivoli Storage Manager, DB2 backups can be stored to other media and local hard disks. With Tivoli Storage Manager, it is possible to maintain a list of backups on tape or optical. Tivoli Storage Manager offers enterprise-wide backup and archive solutions and is also one part of Content Manager.

Tivoli Storage Manager itself is a client-server software, designed to operate over a network. The storage devices are attached to the Tivoli Storage Manager server. DB2 needs to have access to the Tivoli Storage Manager client APIs. This can be easily accomplished by installing the Tivoli Storage Manager Backup Archive client. Then, you need to configure your Tivoli Storage Manager server, Tivoli Storage Manager client, and DB2. Refer to “Backing up to Tivoli Storage Manager” on page 40 for more information about how Tivoli Storage Manager operates in this environment. Detailed configuration steps are provided in “Offline backup to Tivoli Storage Manager” on page 72.

After the setup is complete, backing up a database is as easy as backing up a database to a local file system. It is only necessary to replace the to <directory> clause with use TSM. All other considerations shown in “Full database backup: Offline” on page 22 and “Full database backup: Online” on page 24 are valid for Tivoli Storage Manager, too.

An offline backup using Tivoli Storage Manager is created by the command:

```
db2 backup database <database name> use TSM
```

An online backup using Tivoli Storage Manager is created with the command:

```
db2 backup database <database name> ONLINE use TSM
```

Maintenance of database backup copies

Each time DB2 creates a DB2 backup image into Tivoli Storage Manager, it uses a new unique file name. This does not allow Tivoli Storage Manager to automatically handle keeping a specified amount of database backup versions using backup copy group policies. For this and other reasons, DB2 provides a command line utility called **db2adut1** that uses the Tivoli Storage Manager API to allow you to browse and manipulate the backup images available in the Tivoli Storage Manager server. Given that Tivoli Storage Manager will never erase a backup image of a file that does not have a newer version or has not been explicitly deleted by the client API that stored it, you need to use this tool to do housekeeping to remove old backup images.

As an example, you can delete all full database backups from Tivoli Storage Manager that are older than 60 days with the command:

```
db2adut1 delete full nonincremental older than 60 days
```

This must be run as a regular procedure, keeping in mind that it must be coherent with the policies specified for archived logs retention periods (see “Using user exit” on page 20) and Resource Manager storage area retention periods (see “Server policies for storage area backup” on page 42).

For this backup and maintenance procedure to work properly, you need to set up the correct parameters in the appropriate backup copy group. Table 2-2 shows the required settings.

Table 2-2 Backup copy group settings for DB2 backup images

Setting	Value
Versions Data Exists	1
Versions Data Deleted	0
Retain Extra Versions	0
Retain Only Version	0

Incremental backup

DB2 provides backup facilities for data that has changed since the last offline or online database backup. This will save tape or disk space, but may or may not reduce the time to do a backup, because DB2 still needs to read the data blocks to determine if they have changed since the last backup. When recovery is needed, the database backup and incremental backups are both required to fully recover the database. This can take more time to recover a database. Incremental backups are useful when saving space or when saving bandwidth when backing up over the network.

Important: An incremental backup is a database backup that contains only the changes since the last full database backup.

To enable the use of incremental backups, the DB2 database configuration parameter TRACKMOD must be set to on. To activate this parameter, a restart of DB2 might be necessary. After that, a full database backup is required to initialize the chain of backups. This can be an offline or online backup.

A typical sequence of steps is shown in Example 2-3 on page 27.

Example 2-3 Sequence of setup TRACKMOD and backup using Storage Manager

```
db2 => update database cfg for ICMNLSDB using TRACKMOD ON
DB20000I The UPDATE DATABASE CONFIGURATION command completed successfully.
DB21026I For most configuration parameters, all applications must disconnect
from this database before the changes become effective.
db2 => backup database ICMNLSDB use tsm
Backup successful. The timestamp for this backup image is : 20031018121215
```

Create the incremental backup using Tivoli Storage Manager with the command:

```
db2 backup database ICMNLSDB INCREMENTAL use tsm
```

Figure 2-6 shows a typical incremental backup strategy. Periodically, for example, every Sunday, full backups are created. Changes during the week are backed up with incremental backups. To restore a backup, you need the latest full backup and the latest incremental backup.

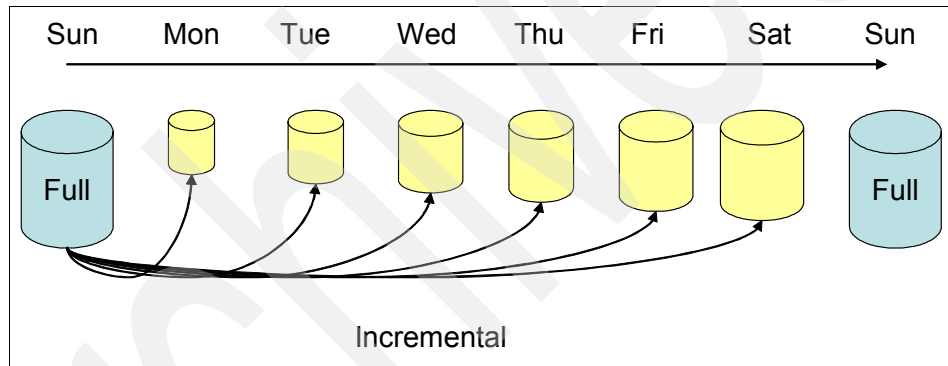


Figure 2-6 Typical incremental backup strategy

For detailed steps about restoring an incremental database backup, refer to the redbook *Backing Up DB2 Using Tivoli Storage Manager*, SG24-6247.

Delta backup (differential)

A delta backup is very similar to an incremental backup. Delta backups are used to save tape or disk space, but may or may not reduce the time to do a backup, too. When recovery is needed, the database backup and *all* delta backups are both required to fully recover the database.

Important: A delta backup is a database backup that contains only the changes since the last backup of every type (full, incremental, or delta).

To enable creation of delta backups, the same steps as described for incremental backups are necessary. To create an delta backup, for example, to a local directory, run:

```
db2 backup database <database name> INCREMENTAL DELTA to <directory>
```

Figure 2-7 shows a typical delta backup strategy. Periodically, for example, every Sunday, full backups are created. Changes during the week are backed up with delta backups. To restore a backup, you need the latest full backup and all delta backups to the restore point.

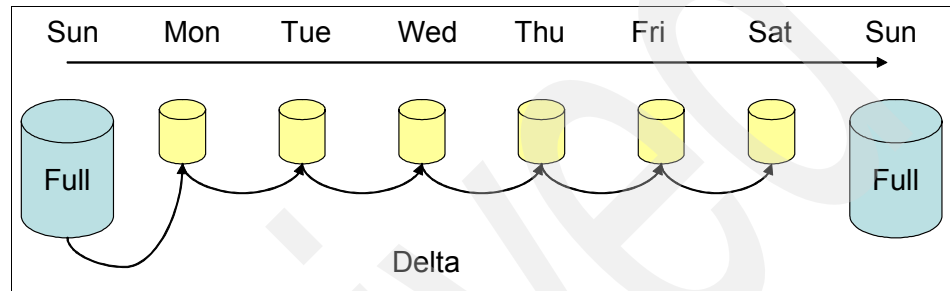


Figure 2-7 Typical delta backup strategy

For detailed steps about restoring a delta database backup, refer to the redbook *Backing Up DB2 Using Tivoli Storage Manager*, SG24-6247.

Tablespace backup

DB2 also offers a feature to back up only some parts of a database. DB2 can create separate backups for every tablespace. This feature is typically used whenever data in different tablespaces has different semantics or different backup requirements. Content Manager, by default, creates several tablespaces for the Library Server, for example, one for maintaining all configurations and definitions and another to store the actual data. Content Manager requires integrity between these tablespaces, so we do not recommend using tablespace backups for Content Manager.

2.3.3 Restoring a DB2 database

Even before creating a backup of a database, it is important to think about the restore requirements. With the different backup options shown in 2.3.2, “Backing up a DB2 database” on page 21 there are different ways to restore a database, mainly distinguished in:

- ▶ Time needed to recover the database
- ▶ Point in time to recover the database to

Restoring a full database backup is the fastest option. By reapplying database log files, the time needed will increase, but you will be able to recover to a later point in time. Your strategy truly depends on your requirements. Especially in very large installations, a mixture of full, incremental, and delta backups, plus logretain, is used.

Restore an image using circular logging

To restore a database that has been created as an offline backup and with circular logging, use:

```
db2 restore database <db name> from <directory>
```

If you need to include the time stamp of the backup, use:

```
db2 restore database <db name> from <directory> taken at <timestamp>
```

If your backup is stored on Tivoli Storage Manager, use:

```
db2 restore database <db name> taken at <timestamp> use TSM
```

Restore an image using archival logging

To restore a database from a backup that has been created as an offline backup with archival logging and without rolling forward the database to a later point in time, use:

```
db2 restore database <db name> from <directory> without rolling forward
```

To restore a database from a backup, which has been created as an offline backup and with archival logging and rolling forward the database to a later point in time using archived log files, use:

```
db2 restore database <db name> from <directory>
```

```
db2 rollforward database <db name> to <timestamp> and complete
```

Note, after having the database restored, it is in rollforward pending mode. With the **rollforward** command, you reapply the log files, and with the **complete** keyword, you change the database status to consistent. You can also roll forward the database to the end the log files using the command:

```
db2 rollforward database <db name> to end of logs and complete
```

2.4 Resource Manager backup and recovery

This section discusses backup considerations for content stored on the Resource Managers. The Resource Manager stores describing data in a DB2 database, which should be backed up just as every other DB2 database, and several other data areas including local hard disks, Tivoli Storage Manager, and VideoCharger.

The Resource Manager consists of a Web application that is deployed to a WebSphere Application Server and several other services that are introduced in this section.

2.4.1 Resource Manager data areas

The Resource Manager typically stores the content of items as files to a local file system, which over time might get migrated to Tivoli Storage Manager. Streamable files, such as audio and video content, can be stored to VideoCharger and migrated to the Multimedia Archive. This section contains general considerations for these data areas. Figure 2-8 on page 33 shows the different data areas and their associated Resource Manager services.

Storage area (LBOSDATA)

The storage area is the main storage location for newly imported documents. It is also the only permanent storage area that is directly handled by Content Manager. Content remains in the storage area until it gets migrated because of an associated migration policy. Content Manager can use several storage areas on separate drives (Windows) or separate logical volumes (UNIX). The name of a subdirectory Content Manager creates for this area is LBOSDATA.

Because content without a secondary copy is only stored in the storage area, this is a very important data area. If you have high requirements to prevent data loss, we strongly recommend that you use mirrored hard disks or redundant disks for the this location.

Tivoli Storage Manager

If objects should be stored to any other media than locally attached hard disks, Content Manager typically migrates these objects to Tivoli Storage Manager. Tivoli Storage Manager is the only software storage management product supported by Content Manager. It offers access to a wide range of storage devices, including, but not limited to, hard disks, optical jukeboxes, and tape libraries.

For Content Manager, Tivoli Storage Manager is like a black box. Content Manager moves the object to a Tivoli Storage Manager management class, a backup copy group. What happens to the object within Tivoli Storage Manager is unknown to Content Manager. Given the objects are stored using the Tivoli Storage Manager API, it is necessary to have the correct settings for the backup copy group to avoid having unnecessary information stored into Tivoli Storage Manager. These settings, which are the same needed for DB2 backup images, are listed in Table 2-2 on page 26. More information about how Tivoli Storage Manager operates on this environment can be found in “Backing up to Tivoli Storage Manager” on page 40.

Tivoli Storage Manager offers a wide range of backup, recovery, and high availability options. For example, Tivoli Storage Manager can create multiple copies of its database and also multiple copies of objects. To find out more about Tivoli Storage Manager options, refer to:

<http://www.ibm.com/software/tivoli/products/storage-mgr/>

Staging area

The staging area is a caching area for Content Manager. Objects that have been migrated to Tivoli Storage Manager are cached during retrieval to the staging area. It is also used for LAN caching. For every object stored in the staging area, there is always a second copy present in the system, either in the storage area or within Tivoli Storage Manager. Because of this, it is not necessary to back up the staging area. Content Manager also maintains information about objects in the staging area, which has to be reset after a restore of the associated Resource Manager database. This information is stored in the table RMOBJECTS on each Resource Manager database. Table 2-3 contains all valid options for the object status column OBJ_STATUS.

Table 2-3 Object status information

Value in OBJ_STATUS	Description
"A"	Resides on a SMS volume.
"B"	Resides on a SMS volume and one on a cache volume (staging or VideoCharger).
"C"	LAN cache pending.
"D"	Ready to be deleted from a SMS volume.
"G"	Staged discarded.
"L"	LAN cached object.
"P"	Delete pending.
"R"	Store pending.
"S"	Resides in staging.
"U"	Has been updated. Update is not complete. Old and new objects exist.

After a restore of a Resource Manager, which includes the database and the storage areas, the information in the database is incorrect. To make sure the objects get restaged with the next request, run the command:

```
db2 update RMOBJECTS set OBJ_STATUS='A' where OBJ_STATUS='B'
```

VideoCharger and Multimedia Archive

Streamable media, typically video and audio files, can be stored to VideoCharger. Based on the MIME type and definitions on the item type, Content Manager will notify VideoCharger if new objects should be imported. The default retrieve operation will only retrieve a meta file that launches the video player. It is also possible to download the object again.

To offload the typically large video and audio files to other media than hard disk, Content Manager offers the Multimedia Archive as extension to VideoCharger. This basically works as a migration to Tivoli Storage Manager, but for a play request the video is restored to VideoCharger.

To learn more about VideoCharger, including how to back up a VideoCharger configuration, refer to the redbook *Content Manager VideoCharger Installation and Integration for Multiplatforms*, SG24-6410.

2.4.2 Resource Manager services

The Content Manager Resource Manager does not only consist of the Web application that is running within WebSphere and serving import and retrieve requests. Four background services that can be started independently from the Web application also belong to the Resource Manager.

Figure 2-8 on page 33 shows the Resource Manager services and the data areas on which they are working. The upper half of the figure shows services related to VideoCharger, the lower half for non-streamed objects. Not shown is the replicator service that will typically work between two Resource Managers.

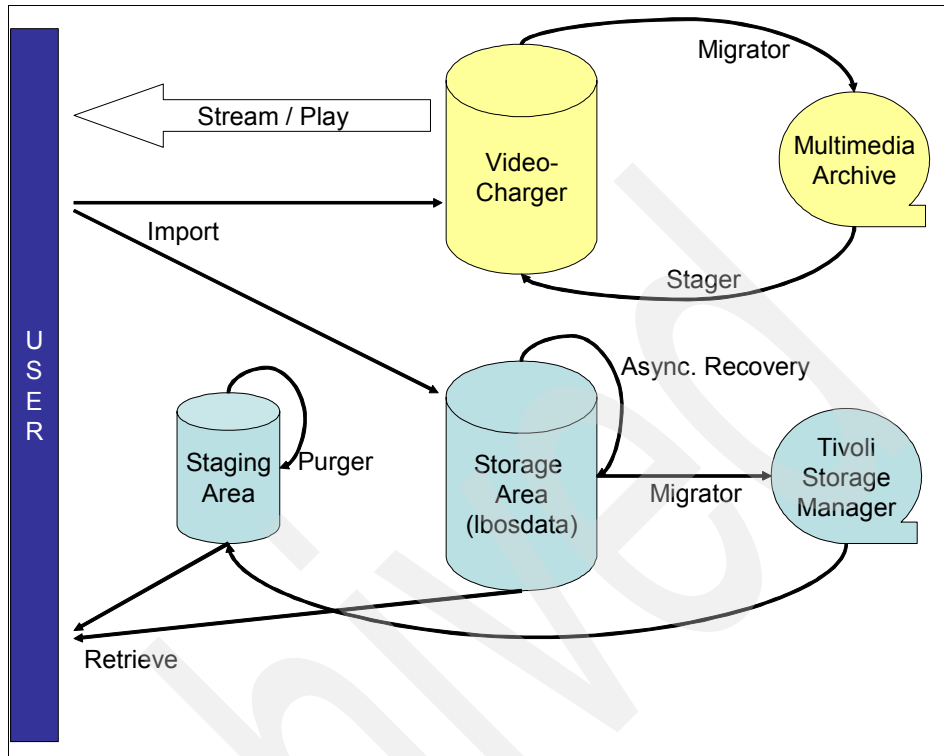


Figure 2-8 Resource Manager data areas and services

The four main services, purger, stager, migrator, and replicator, can be started separately from the Resource Manager Web application. On Windows, these four applications are Windows services and can be started and stopped using the Windows Services Control Panel, as shown in Figure 2-9 on page 34, or using the `net start <name of service>` command in batch operations.

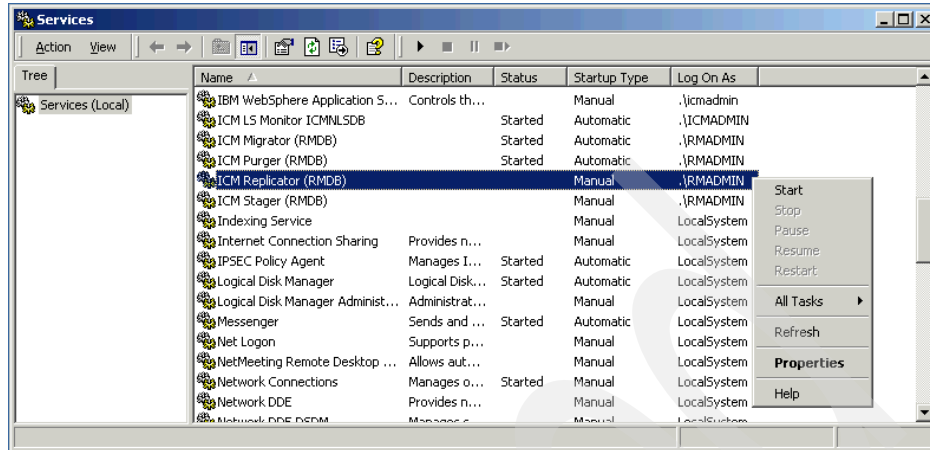


Figure 2-9 Resource Manager services on Windows

On UNIX, these services can be started using the script `rc.cmrmproc -act start` and stopped using the `rc.cmrmproc -act stop` script provided by Content Manager, as shown in Example 2-4.

Example 2-4 Starting Resource Manager services on UNIX

```
root@cm44/etc# ./rc.cmrmproc -act start
PROCACTION: start
2003-11-10 22:56:22,441 ICM0000: Licensed Materials - Property of IBM
IBM Content Manager for Multiplatforms V8.2 (program number 5724-B19)
(c ) Copyright IBM Corp. 1994, 2002, 2003. All Rights Reserved.
US Government Users Restricted Rights - Use, duplication or disclosure
restricted by GSA ADP Schedule Contract with IBM Corporation
RMMigrator --> RMDB
RMPurger --> RMDB
RMReplica --> RMDB
RMStager --> RMDB

# root@cm44/etc# ./rc.cmrmproc -act stop
PROCACTION: stop
2003-11-10 23:06:00,483 ICM0000: Licensed Materials - Property of IBM
IBM Content Manager for Multiplatforms V8.2 (program number 5724-B19)
(c ) Copyright IBM Corp. 1994, 2002, 2003. All Rights Reserved.
US Government Users Restricted Rights - Use, duplication or disclosure
restricted by GSA ADP Schedule Contract with IBM Corporation
RMMigrator --> RMDB
input: cm44 7500 shutdown
sending shutdown to process
client waiting for status info
client got down
```



```
RMPurger --> RMDB
input: cm44 7501 shutdown
sending shutdown to process
client waiting for status info
client got down
RMReplica --> RMDB
input: cm44 7502 shutdown
sending shutdown to process
client waiting for status info
client got down
RMStager --> RMDB
input: cm44 7503 shutdown
sending shutdown to process
client waiting for status info
client got down
```

Purger

The job of the purger process is to maintain the staging area size. The staging area is used to cache objects from Tivoli Storage Manager and remote Resource Managers (for LAN cache). For example, when a client makes a request for a document that is stored in an optical jukebox managed by Tivoli Storage Manager, the document is first copied to the staging directory and then sent to the client. Any future requests for this particular document will be much faster, because the document would then be on the hard disk, and not on a slower optical platter.

When the staging area size reaches a preset upper limit, the purger will begin to remove files until it reaches a preset lower limit. For example, Figure 2-10 on page 36 depicts a 200 MB staging area, which has an upper threshold of 190 MB and a lower threshold of 10 MB. Whenever the staging area reaches 190 MB in size, the purger process will begin to selectively delete files from the staging directory, and will continue to do so until the directory reaches 10 MB in size.

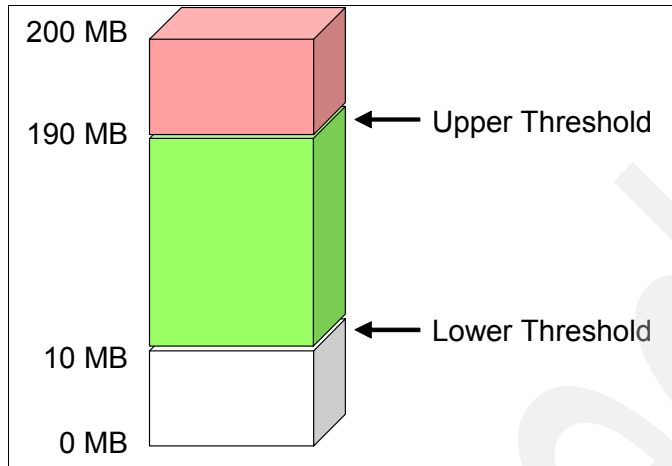


Figure 2-10 Staging area thresholds

Figure 2-11 shows how to accomplish these settings in the Content Manager System Administration Client. To get this panel, click **Resource Managers** → **RMDB** → **Staging Area**.

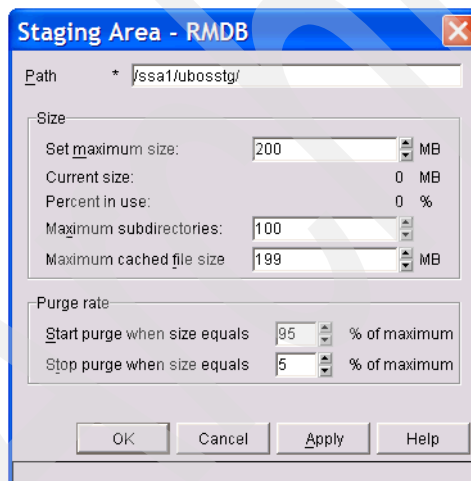


Figure 2-11 Staging Area properties

Stager

The job of the stager process is to delete staged objects from a VideoCharger server. When using a Media Archive Server with VideoCharger, videos that are archived on the Media Archive Server cannot be directly streamed to clients. When a play request for an archived video is received, the video object must be

temporarily staged (or cached) to the VideoCharger server. When the VideoCharger stager gets too full, the stager process will delete some, or all, of the staged objects.

Migrator

The job of the migrator is to move documents between storage classes and to remove deleted documents from the database. The migrator process is used to execute the rules set forth by the migration policies. If you forget to start the migrator process, documents will never be moved.

The second job of the migrator process is to remove objects from the Resource Manager server that have been marked for deletion by the asynchronous recovery utilities. When a client deletes a document, the database entries are immediately removed from the Library Server, but still remain on the Resource Manager (for performance reasons). The asynchronous recovery tools will mark the objects in the Resource Manager database that need to be removed. When the migrator starts, it will perform the deletion of these documents from the database. Therefore, even though you might not be planning to migrate documents between storage classes, you should always ensure that the migrator process has been started so that deleted documents can be removed from the database tables.

Replicator

The replicator creates additional copies of documents stored to collections enabled for replication. When defining a collection, you can choose a copy destination on the same or a different Resource Manager. The replicator will copy these documents asynchronously, for example, to a secondary Resource Manager, as shown in Figure 2-12 on page 38.

The schedule and frequency used to run the replicator highly depends on the requirements. Frequent runs all over the day make sure that the secondary collections are up-to-date; however, they have a large impact on performance. A single run in off-peak hours reduces hardware and bandwidth requirements, but increases the risk of having not yet replicated documents unavailable.

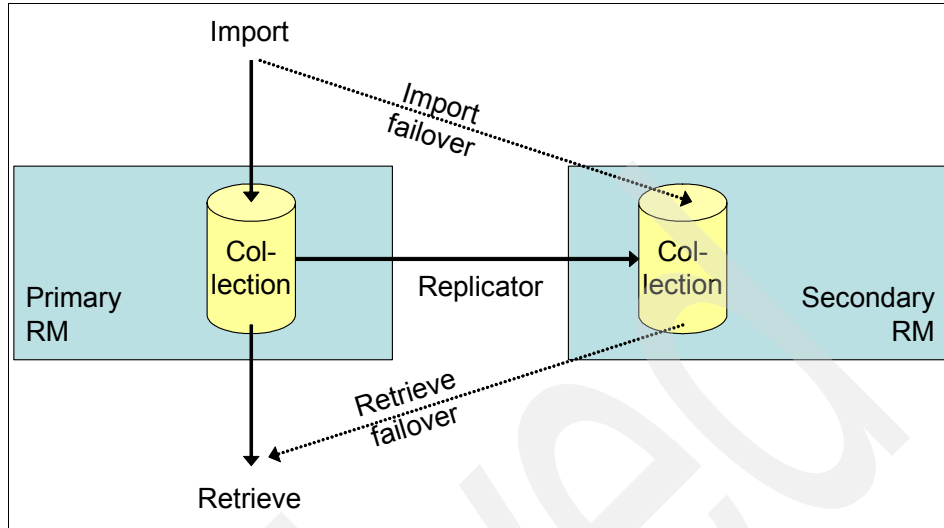


Figure 2-12 Content Manager replication

Content Manager replication is not load balancing. As long as the primary Resource Manager is available, all import and retrieval operations will be managed by it. Only if the primary Resource Manager goes down does the Library Server monitor switch to secondary or other standby systems.

Asynchronous recovery utilities

The asynchronous recovery utilities consist of two programs used for document deletion and transaction reconciliation. These utilities are used to maintain and to restore the consistency between the Library Server and Resource Manager databases.

The *document deletion utility* will determine which documents have been deleted from the Library Server database and will then flag these documents in the Resource Manager databases for deletion. The migrator will actually perform the removal of database entries for deleted documents.

The *transaction reconciliation utility* is used to roll back any failed transaction and ensures that the Resource Manager database is in sync with the Library Server database.

These two utilities are automatically executed before the migrator performs any work. However, you do have the option of manually running these utilities yourself.

To manually run these utilities on Windows, open a DB2 command window, using **db2cmd**. To start the deletion reconciliation utility, run **icmrmdel.bat** (located by default in C:\Program Files\IBM\CM82\bin). To start the transaction reconciliation utility, run **icmrmtx.bat** (located by default in C:\Program Files\IBM\CM82\bin).

To run these utilities on AIX, log on as the Resource Manager administrator (by default rmadmin). To start the deletion reconciliation utility, run **icmrmdel.sh** located in /usr/lpp/icm/bin. To start the transaction reconciliation utility, run **icmrmtx.sh** located in /usr/lpp/icm/bin.

We recommend that you run the asynchronous recovery utilities before performing a backup of your system. This not only ensures your databases are in a consistent state, but also removes any database entries that represent deleted documents.

2.4.3 Backup and recovery of the storage area

The storage area represents the most important data location for a Content Manager Resource Manager. Imported documents are stored to a storage area based on associated collections. If replication is not used, there is no second copy of this data available. Because of this, we highly recommend placing the storage area on a mirrored or redundant disk.

When creating the Resource Manager database, a default storage area is created. Within the Content Manager System Administration Client, you can review the current settings in the storage systems section. Note that there might be more than one file system storage system defined.

A storage system specifies the location, or volume, of where an object is stored and is directly associated with a storage class. A file system volume represents a physical or logical partition of a hard disk. For example, on Windows, a file system volume would be represented by a drive label (that is, C_Drive). On AIX, a file system volume would be represented by a file system mount point (that is, /home/rmadmin/lbosdata).

When Content Manager uses the storage area, it creates a configurable number of subdirectories to enable an efficient and fast object storage and retrieval. Within these subdirectories, Content Manager will store the objects as files, using internal Content Manager structures for the file names.

Files will be removed from the storage areas by the deletion reconciliation utility or by the migrator process. They get updated if a user modifies a Content Manager document and versioning is not used.

Backing up to a file system

Because the storage area is a standard file system, you can back it up using standard backup tools. This can be as easy as copying the files using the copy command of the operating system or using the built-in backup utilities of your operating system.

Backing up to Tivoli Storage Manager

Using Tivoli Storage Manager to manage the storage area backups, as well as a secondary storage device, is a good option and a very common practice. This approach lets you take advantage of the incremental backup capabilities and storage management of Tivoli Storage Manager to handle the high volume of information usually available in any Content Manager installation.

Note, all the following concepts mentioned are explained in more detail in *IBM Tivoli Storage Manager Administrator's Guide*, GC32-0782.

Important: This guide is not intended to provide all the knowledge necessary to become an expert in Tivoli Storage Manager backup implementation. Include your Tivoli Storage Manager specialist when you plan for your Content Manager system backups.

Tivoli Storage Manager is an enterprise network storage and backup manager. There is a server component which provides various backup and archive services and holds all the files stored in the system. Client components participating in the system give their files to the Tivoli Storage Manager server across the network. These client components include the Tivoli Storage Manager backup-archive client, the Tivoli Storage Manager API used by the Content Manager Resource Managers, Tivoli Storage Manager administrative interfaces, and others.

The Tivoli Storage Manager backup-archive client is the component used to back up the files in a server in the network. In particular, this is the client that needs to be used to back up the files in the storage area of the Resource Manager. For the Content Manager Resource Manager secondary storage to Tivoli Storage Manager and for DB2 to back up directly to Tivoli Storage Manager, we need to use the Tivoli Storage Manager client API. Usually, both client pieces come together as a single installation package.

How incremental file system backup works

Tivoli Storage Manager provides a unique, efficient method for its standard backups such as the Content Manager Resource Manager storage area backup. This method, called progressive incremental backup, is the default recommended

method. Tivoli Storage Manager also provides other methods that are not covered here.

One very important characteristic of Tivoli Storage Manager is that the server manages the files it stores using a relational database to know the exact location, status, expiration, and other information of each file. This way, the task of remembering which tape volume holds each file is relieved from the backup administrator.

Using this capability, Tivoli Storage Manager takes the incremental backup concept introduced in 2.3.2, “Backing up a DB2 database” on page 21 one step further. After the initial full backup of a client, no additional full backups are necessary. The server database keeps track of which files need to be backed up, and when necessary, entire files are backed up. This way when a restore is needed, the administrator just specifies the file system to be restored from a backup-archive client, and the Tivoli Storage Manager takes care of mounting the appropriate volumes to provide all the missing files.

This process is further improved by the concepts of collocation and space reclamation for tape volumes. Before any file can be stored into Tivoli Storage Manager, a storage pool must be defined. This is no more than a named group of storage volumes such as tapes that will hold files from specified clients.

As time goes by and you keep doing incremental backups of the storage area, new tapes will need to be used. Nevertheless, there will be files that will be deleted from the storage area, because they were moved to secondary storage, or because they were deleted or expired from Content Manager. The backup copy for these files will also expire in the Tivoli Storage Manager server, leaving empty spaces in random places of the tape volumes. For this reason, Tivoli Storage Manager runs a periodic process called space reclamation that takes several tapes with enough empty space, move their remaining content to new ones, and leaves the remaining for reuse. This way, the amount of tapes that hold backup information remains more or less static and makes the restore process easier. This process is depicted in Figure 2-13 on page 42.

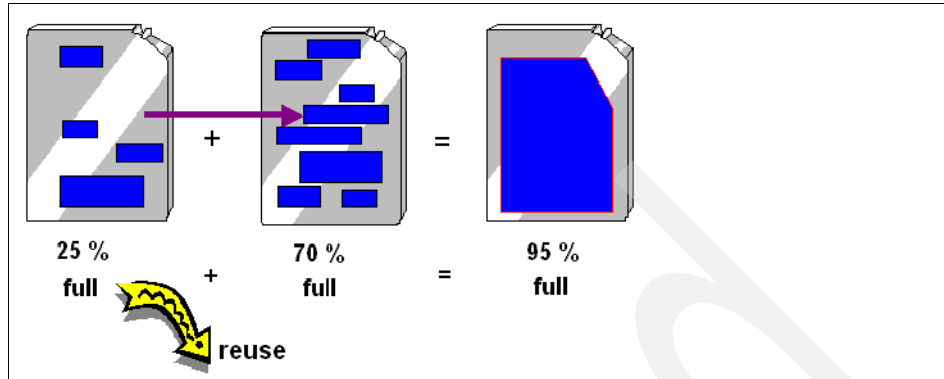


Figure 2-13 Tivoli Storage Manager space reclamation

Conversely, collocation tells Tivoli Storage Manager to use a different set of volumes for each client node that is storing data. This way, the amount of tape volumes needed for the restore of a particular Resource Manager backup is even lower.

There are several other operations a client can perform to store and manipulate its data into a Tivoli Storage Manager server. These include backup-restore, archive-retrieve, and space management. Also, Tivoli Storage Manager clients can use other methods for backup such as full backups, differential backups, or backup set creation. These are not covered here nor recommended for Content Manager Resource Manager storage area backup.

Server policies for storage area backup

As part of the policy domain configuration for the Content Manager client nodes, we recommend using the following guidelines to set up the backup copy group assigned to the management class used to hold Resource Manager storage area file system backups.

The main goal of backing up the storage area is to be able to restore the Resource Manager at least to the latest possible time before a problem occurs. To be able to restore a Resource Manager, you need to have *all* the files that existed at certain point in time in the storage area. Doing a partial restore is very problematic and can lead to inconsistency states in the Content Manager system that can be very hard to recover.

For this reason, we recommend using a SHAREDYNAMIC value for the Copy Serialization setting to force Tivoli Storage Manager to copy all the files even if they are currently in use. Nevertheless, it is very unlikely that a file will be modified during backup time even during online backups.

For this same reason, it is also important not to limit the amount of backup copies to keep in Tivoli Storage Manager according to the number of file versions. If you want to be able to go back to a storage area image to a certain number of days in the past, you should specify this value in both the Retain Extra Versions and Retain Only Versions fields. Then, no matter how many modifications exist for a particular stored object, you will have all the versions as they existed at least for the amount of time specified. Otherwise, if a particular object was deleted because the amount of versions was exceeded, this object will be missing if you do a restore to a previous point in time. Table 2-4 summarizes this idea for a sample configuration that holds backup images of the storage area for 60 days.

Table 2-4 Sample Tivoli Storage Manager copy group configuration

Setting	Value
Versions Data Exists	NOLIMIT
Versions Data Deleted	NOLIMIT
Retain Extra Version	60
Retain Only Version	60
Copy Serialization	SHAREDYNAMIC

Keep in mind that these settings must be coherent with the policies used for database backup images deletion (see “Maintenance of database backup copies” on page 25) and archived logs retention periods (see “Using user exit” on page 20).

Client configuration concepts

To be able to back up and restore a Resource Manager storage area to Tivoli Storage Manager, you need to install the Tivoli Storage Manager backup-archive client in the Resource Manager node. This component needs to be configured to connect the Tivoli Storage Manager server, to back up the right files when asked, and to use the correct options.

The Tivoli Storage Manager client takes its configuration from an options file. This option file is identified by an environment variable named DSM_CONFIG. In Windows platforms, it consists only of this single file, usually called dsm.opt and located inside the Tivoli Storage Manager backup-archive client program directory structure. In AIX, Sun Solaris, and Linux, there is also a system-wide client option file named dsm.sys that holds some of the configuration variables and applies to all Tivoli Storage Manager client instances.

Inside this Tivoli Storage Manager file, you configure the connection to the Tivoli Storage Manager server, the names of the files that should be backed up or skipped, and any possible configuration parameter available from the client side.

For details about these configuration parameters, refer to *IBM Tivoli Storage Manager for Windows Backup-Archive Clients Installation and User's Guide*, GC32-0788 or to *IBM Tivoli Storage Manager for UNIX Backup-Archive Clients Installation and User's Guide*, GC32-0789.

Initiating backup operations

After your Tivoli Storage Manager client and server are configured to back up the Resource Manager storage area, you only need to decide how you can include this backup in the context of the Content Manager system backup, along with all the other components.

There are several ways you can initiate a Tivoli Storage Manager file system backup, but we concentrate on the two main different approaches: scheduled in the Tivoli Storage Manager server or started from the client side.

Tivoli Storage Manager includes a central scheduler which can be used to specify, in a single place, when and at what time each system in your network must be backed up. We recommend using this option to back up your Content Manager system if you are using Tivoli Storage Manager as your corporate network backup system.

In this scenario, we recommend scheduling a normal backup operation and using the `preschedulecmd` and `postschedulecmd` options in the client options file to include the rest of the Content Manager specific backup operations such as server pause and resume or DB2 database backup.

The use of the Tivoli Storage Manager scheduler could be handy in the case of a multinode Content Manager system. The idea is to create a specific schedule for the Content Manager backup, associate all the Content Manager client nodes to it, and specify a short time window to force all backup operations to take place at the same time. We also recommend that you use server prompted scheduling for these kind of operations. In any case, take into account that a specific scheduler component must be configured in the client for the schedule to take place. Once more, refer to the Tivoli Storage Manager documentation for details.

If you decide to use another scheduling facility and want to start a backup to Tivoli Storage Manager from an external source, you can use the backup-archive client command line interface. This is a program that includes all the functions of the regular Tivoli Storage Manager backup-archive client from the command prompt of the operating system.

This interface can run in batch mode, allowing entire operations to be specified from other scripts or programs. Batch mode is used whenever you call the command line executable followed by any Tivoli Storage Manager commands. For example, the following command performs a default, non-progressive,

incremental backup of the files configured in the options file in a Windows system.

```
C:\> "C:\Program Files\Tivoli\TSM\baclient\dsmc.exe" incremental
```

When you prepare the scripts for your Content Manager system backup, you will most probably include this or a similar line as part of other automatically run scripts. In that case, remember to either specify the password as part of the command or to configure the client to store and generate passwords with the **passwordaccess generate** option. Otherwise, the command line client will wait for a password, halting the entire backup process.

2.5 Content Manager configuration

Content Manager uses some configuration files that can change over time. Table 2-5 shows the default location for these configuration files and Table 2-6 on page 46 shows the most important configuration files.

Table 2-5 Configuration file default locations

Location	Windows	AIX	Sun Solaris	Linux
%ICMROOT%	C:\Program Files\IBM\CM82	/usr/lpp/icm	/opt/IBMicm	/opt/IBMicm
%CMBROOT%	C:\CMBROOT	/usr/lpp/cmb	/opt/IBMcmb	/opt/IBMcmb
%WASROOT%	C:\Program Files\IBM\WebSphere\AppServer	/usr/WebSphere/AppServer	/opt/WebSphere/AppServer	/opt/WebSphere/AppServer
%CMCOMMON%	C:\Program Files\IBM\Cmgmt	/usr/lpp/cmb/cmgmt	/opt/IBMcmb/cmgmt	/opt/IBMcmb/cmgmt
%HTTPROOT%	C:\Program Files\IBM\HttpServer	/usr/IBMHttpServer	/opt/IBMHttpServer	/opt/IBMHttpServer

We strongly recommend that you create a backup of the files shown in Table 2-6 on page 46 on a regular basis, at least every time you update your Content Manager configuration. This will help to ensure that you can restore your Content Manager environment quickly even if the program code needs to be reinstalled.

Table 2-6 Configuration files

File name	Location	Component	Purpose
cmbcmenv.properties	%CMCOMMON%	LS and II4C	Points to other config files.
cmbicmenv.ini	%CMCOMMON%	LS	Contains connect user ID and passwords.
cmbicmsrvs.ini	%CMCOMMON%	LS	Contains list of available servers.
cmadmin.properties (Windows and Linux only)	%CMCOMMON%	Admin Client	Settings for Content Manager System Admin Client.
icmrm.properties	%WASROOT%/installedApps/<node name>/icmrm/icmrm.ear/icmrm.war/WEB-INF/classes/com/ibm/mm/icmrm	RM	Resource Manager settings such as user ID, password, and JDBC connection.
dsm.opt	N/A	RM (Tivoli Storage Manager)	Tivoli Storage Manager client node configuration file.
icmrm_<procname>_logging.xml	%WASROOT%/installedApps/<node name>/icmrm/icmrm.ear/	RM	Resource Manager logging configuration.
httpd.conf	%HTTPROOT%/config	RM, eClient (HTTPD)	HTTP server configuration.
key.kdb and key.sth	N/A	RM	SSL key database and password file.
setprocenv.sh (UNIX) setprocenv.bat (Windows)	%ICMROOT%/config	RM	Configuration for Resource Manager services.

2.6 Additional considerations for Content Manager

This section includes additional scripts provided by Content Manager and commands frequently needed to do maintenance of Content Manager components. It also gives an overview of optional components, such as Tivoli Storage Manager for the Resource Manager or full text search.

2.6.1 Stopping Content Manager activity

To create a cold or warm backup, it is necessary to stop activity on Content Manager components. Although this can be easily accomplished by stopping

DB2, WebSphere, and DB2 extenders, there are more effective ways to do this while still maintaining some service availability.

Before performing an offline backup, it is necessary to disconnect all clients from the Library Server, to stop the Library Server monitor, to shut down the Resource Manager Web application, and to stop all Resource Manager services.

Before performing a warm backup, we recommend stopping at least the Resource Manager services and full text indexes updates.

Disconnect all clients

If you need to disconnect all clients from the database, we recommend stopping all client applications. This can be accomplished by ending all client applications or by disconnecting them from the Library Server database. To disconnect a custom client application, you need to use the following Content Manager API calls:

- ▶ Java™ and C++ Object Oriented API:

```
DKDatastoreICM.disconnect();  
DKDatastoreICM.destroy();
```

- ▶ Java Beans:

```
CMBConnection.disconnect();
```

Note: On the Object Oriented API, the `disconnect()` method of the ICM datastore class logs off the user from Content Manager only, still keeping the DB2 connection alive. The `destroy()` method also ends the DB2 connection.

However, many times you do not have the control on all client workstations, so you need to force all connections from the server. To disconnect all applications from a DB2 database such as the Library Server, run the following command:

```
db2 force application all
```

Note: This command forces all DB2 applications attached to the current instance. If there is more than one database active in an instance, you can selectively disconnect these applications as shown in “Full database backup: Offline” on page 22.

Stop the Library Server monitor

Content Manager provides a fail-over service that verifies whether the Resource Managers are available. If you are trying to store objects into a Resource Manager that is unavailable, Content Manager tries to store into the next available Resource Manager. Without this fail-over service, you would get an error if you tried to store objects in a Resource Manager that was unavailable.

The fail-over service monitors the availability of the Resource Managers based on the interval that you set on the Interval to check server availability field in the Library Server Configuration window. For example, if you set 60 seconds as the interval, it checks availability every 60 seconds. This service should remain running. The Library Server monitor service is named ICMPLSAP (Portable Library Server Asynch Process). It opens a connection to the Library Server database.

To stop and start service:

- ▶ On Windows, you can check the status, start, and stop the service from the Services Control Panel, as shown in Figure 2-9 on page 34, or you can use the following commands in batch files:

```
net stop "ICM LS Monitor ICMNLDB"  
net start "ICM LS Monitor ICMNLSDB"
```

Where ICMNLSDB is the name of your Library Server database.

- ▶ On UNIX, you can query the status, start, and stop the Library Service monitor service using:

```
/etc/rc.cmllproc -status    #Query status  
/etc/rc.cmllproc           #Start  
/etc/rc.cmllproc -shutdown #Stop
```

Stop Resource Manager services

To be able to create a consistent backup between the Resource Manager database and its associated storage areas, it is important to stop any activity on the storage areas. Although you can prevent addition and updates of files only by stopping all client activities, you can stop the removal of files by stopping the Resource Manager services: stager, purger, replicator, and migrator. For more details about these components, see 2.4.2, "Resource Manager services" on page 32.

On Windows, the stager, purger, and migrator services all run as a Windows service and can be started or stopped through the Windows Services Control Panel or using the **net stop "<name of service>"** command. On UNIX, these services are run as background processes and are started from the inittab file. To stop these services, run **rc.cmllmproc -act stop**, as shown in Example 2-4 on page 34.

Content Manager pause utility

You can pause the Library Server to stop all new transactions, but allow transactions that are in progress to complete. With no transactions running on the Library Server, there will be no client-initiated actions to any Resource Manager. With the work suspended, you can create a consistent backup of all Content Manager servers.

To initiate the Library Server pause process, run the command line script **pauseserver.bat** or **pauseserver.sh** to set the time at which the pause is to begin. This command updates the **SUSPENDSEVERTIME** field in the **ICMSTSYSCONTROL** table of the Library Server database. When that time is less than or equal to the current time, any new transactions are rejected. If an application is storing an object to a Resource Manager, those operations are allowed to complete if they can do so within the time specified in the **MAXTXDURATION** field in the **ICMSTSYSCONTROL** table. After that time, any request to the Library Server is rejected.

Important: Pausing a Content Manager Library Server stops all new transactions, which includes read-only operations.

To end the pause process, run the command line script **resumeserver.bat** or **resumeserver.sh** to set **SUSPENDSEVERTIME** to null. As an alternative, the **pauseserver** command can be run to set the time to a future value, such as the same time on the following day.

The scripts to pause and resume the servers are located in the **%ICMROOT%\bin** (Windows) or **\$ICMROOT/bin** (UNIX) directory. To use the scripts, complete the following steps:

1. Start a DB2 command window.
2. Change the directory to **%ICMROOT%\bin** (Windows) or **\$ICMROOT/bin** (UNIX).
3. To pause, enter:
`pauseserver.bat <Library Server database name> <userid> <password> <suspend server time>`

The format of the suspend server time is 2002-06-12-19.27.49.000000.

4. To resume, enter:
`resumeserver.bat <Library Server database name><userid><password>`

When the Content Manager Library Server is paused, both the client applications, the Content Manager Client for Windows and eClient, will show an exception message to a user who tries to open a transaction. We recommend that you educate your users about this message, so they continue to work after the server has been resumed. With these two standard applications, users will also not be able to retry failed operations without having to reenter updated information.

If you plan to use the pause server and resume server utilities together with a custom application, we strongly recommend that you implement a handling of exceptions caused by a paused server. Your application needs to take care of two exceptions, as shown in Example 2-5 on page 50. We recommend that you

catch the exception and show a meaningful message to the user. Whenever possible, include a retry function in that message, so the user does not lose any work during the server maintenance.

Example 2-5 Error messages related to a paused Content Manager server

ICM7751 The server has been paused for backup.

Please try again later.

Explanation: The server has been paused for backup. Try again later.

Component: Library server

Error ID: RC_QUIESCE_SERVER_PAUSED

ICM7752 The transaction has exceeded the maximum allowed time and will be terminated as a result of a pause server request.

Explanation: The transaction has exceeded the maximum allowed time and will be terminated as a result of a pause server request.

Component: Library server

Error ID: RC_QUIESCE_TX_MAXDURATION_EXCEEDED

NSE indexes updates

If using the full text search capabilities with Content Manager, you need to provide an update schedule for every index you create. Using the Content Manager System Administration Client, as shown in Figure 2-14 on page 51, you can provide the update frequency in the Index update settings area. What you cannot do with the Content Manager System Administration Client is prevent updates in a special time frame, such as a backup window.

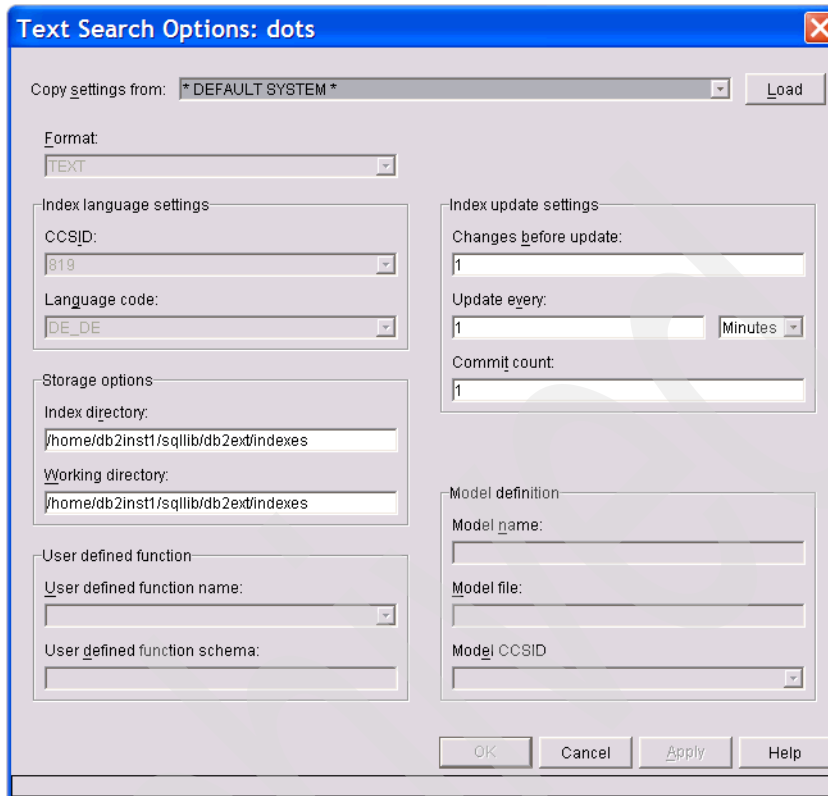


Figure 2-14 Text Search Options window

We recommend that you do not schedule text index updates when running an online backup. For the Library Server and Resource Manager, this primarily affects performance, but for creating a backup of the text indexes itself, it is mandatory to not have any index update operations.

The information about when and how frequently an index is updated is stored in a DB2 table in the Library Server database. It is possible to query and update this information using an SQL command, as shown in Example 2-6.

Example 2-6 Text index update frequency

```
db2 => connect to ICMNLSDb user icmadmin using password
```

Database Connection Information

```
Database server      = DB2/6000 8.1.2
SQL authorization ID = ICMADMIN
Local database alias = ICMNLSDb
```

```
db2 => select UPDATEFREQUENCY from DB2EXT.TEXTINDEXES

UPDATEFREQUENCY

-----
-----
-----
-----
NONE

D(*)H(*)M(0,10,20,30,40,50)

D(*)H(*)M(0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59)

D(*)H(3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22)M(0,10,20,30,40,50)

4 record(s) selected.
```

In this example, there are four text indexes defined. The first index has no scheduled updates. The second index is updated every 10 minutes, every day, every hour. The third index is similar, just having scheduled updates every minute. The last index has settings that are useful for creating a backup of the text index itself. Between 10 p.m. and 3 a.m., there are no updates scheduled; otherwise, the index is updated every 10 minutes. Therefore, between 10 p.m. and 3 a.m. is a perfect time for backing up the index, because there are no operations or only read-only operations on the index files.

2.6.2 Library Server extensions

Besides the Library Server database and its configuration files, there are a few more components associated with the Library Server. This section introduces the components and explains how to back up and restore them.

Access modules

The Content Manager Library Server uses compiled code, the access modules, to allow static queries to be performed. Static queries, as opposed to dynamic queries, should reduce the amount of time needed to perform a search.

To be able to build the modules, the Library Server needs access to a C compiler and the DB2 application development toolkit. All access modules are stored in a single directory, as shown in Table 2-7 on page 53.

Table 2-7 Access module location

Operating system	Access module location
Windows	C:\Program Files\IBM\CM82\<LS DB name>\DLL
AIX	~db2fenc1/<LS DB name>/DLL
Sun Solaris	~db2fenc1/<LS DB name>/DLL
Linux	N/A

We recommend that you back up these files every time you create a backup of the Library Server database, or at least every time you modify a Content Manager item type. If the access modules must be recreated, use the tool **TRebuildCompTypeICM**, which can be found in the %ICMROOT%/config directory. Example 2-7 shows a sample use and result of this command.

Example 2-7 Rebuilding Library Server access modules

```
root@cm44 /usr/lpp/icm/config# java TRebuildCompTypeICM ICMNLSDB icmadmin
password icmadmin log.log
Number of views found: 15
Generating access module for view with ID: 200
Generating access module for view with ID: 201
Generating access module for view with ID: 202
Generating access module for view with ID: 203
Generating access module for view with ID: 204
Generating access module for view with ID: 205
Generating access module for view with ID: 206
Generating access module for view with ID: 207
Generating access module for view with ID: 208
Generating access module for view with ID: 300
Generating access module for view with ID: 301
Generating access module for view with ID: 302
Generating access module for view with ID: 303
Generating access module for view with ID: 304
Generating access module for view with ID: 400
All access modules rebuilt
```

Some releases of Content Manager stopped the usage of access modules. In the current Content Manager Express and Content Manager for Linux versions, access modules are no longer used.

NSE indexes

IBM DB2 UDB Net Search Extender (NSE) is required to provide text search capabilities for your content management system. It allows clients to perform text searches on both metadata and document contents. NSE must be installed on the Library Server workstation.

When creating a full text index, NSE prompts you for a location of the index and work files. An example is shown in Figure 2-14 on page 51. The values are provided in the Storage options area. If no values are provided, the text indexes and work files are stored to a default location that is kept in the table DB2EXT.DBDEFAULTS in the Library Server database. Example 2-8 shows sample values for an AIX system.

Example 2-8 Net Search Extender default values

db2 => connect to ICMNLSDB user icmadmin using password

Database Connection Information

Database server = DB2/6000 8.1.2
SQL authorization ID = ICMADMIN
Local database alias = ICMNLSDB

db2 => select * from DB2EXT.DBDEFAULTS

DEFAULTNAME	DEFAULTVALUE

CCSID	819
LANGUAGE	EN_US
FORMAT	TEXT
MODELCCSID	819
UPDATEFREQUENCY	NONE
UPDATEMINIMUM	1
INDEXDIRECTORY	/home/db2inst1/sqllib/db2ext/indexes
WORKDIRECTORY	/home/db2inst1/sqllib/db2ext/indexes
UPDATECOMMITCOUNT	0
CLEARCOMMITCOUNT	0
CACHEDIRECTORY	/home/db2inst1/sqllib/db2ext/indexes
PCTFREE	50
USEPERSISTENTCACHE	1
AUTOMATICREORG	1
TREATNUMBERSASWORDS	0
INDEXSTOPWORDS	1
UPDATEDELAY	0
VERSION	NSE V8.1.2

18 record(s) selected.

Backing up a full text index is done by backing up the files in the index and work directory. You should create a backup of the Library Server database at the same time, because the NSE will keep track of already indexed objects in this database.

2.6.3 Resource Manager extensions

The Resource Manager mainly stores all its data to a storage area or Tivoli Storage Manager and maintains this files using a DB2 database. Because especially the storage area might grow quickly and reach sizes that are impractical for doing a full or incremental backups, there are several other options available to reduce the risk of data loss.

Resource Manager replication

For enhanced security and ability to retrieve, you can replicate object data from a primary Resource Manager to a replica Resource Manager (also known as a backup Resource Manager). The replica Resource Manager is then available for retrieval and update in case the primary Resource Manager is unavailable. You can define your options for replication when you define a Resource Manager configuration in the New Resource Manager Configuration window of the System Administration Client. On the Replicator Schedule page (the Replicator Schedule tab on the window), you can define the replicator schedule to specify when you want the replicator to run. On the Cycles page (the Cycles tab on the window), you can set the amount of time before the system checks to see if replication is necessary.

Replication is not intended to replace normal system backups. It is only an additional tool to ease recovery from hardware failures and other such events. We recommend running the replicator during times when there is little server activity. For more details about the replicator service, which performs the copy of the documents, see “Replicator” on page 37.

2.6.4 Validation utilities

The purpose of the validation utilities is to analyze discrepancies between three components: the Library Server, the Resource Manager, and the storage systems used by the Resource Manager through its defined device managers. Any of these components could fail and require a restoration through a backup that could be out of sync with the other two components.

Because there is no direct link between the Library Server and the storage system (an example of a storage system could be VideoCharger or Tivoli Storage Manager), differences must be reported between the Library Server and the Resource Manager and the Resource Manager and the storage system.

The *Resource Manager/Library Server validation utility* generates reports that describe discrepancies between the Library Server and the Resource Manager. The *Resource Manager/volume validation utility* provides reports on discrepancies between the Resource Manager and the storage system.

The reports are in XML. You can use commonly available XML tools or browsers to view or manipulate the utility output files. Content Manager installs the XML DTD required by the validation utility output files.

The shell scripts and batch files that invoke the validation utilities are located in the bin directory in the Resource Manager installation directory.

Run the Resource Manager/Library Server validation utility with the command:

```
icmrmlsval.sh OR icmrmlsval.bat
```

Run the Resource Manager/volume validation utility with the command:

```
icrmrvolval.sh OR icrmrvolval.bat
```

The validation utility creates and drops a temporary DB2 table. The environment script requires the Resource Manager database user ID, password, schema, Web application path, and DB2 instance. To set the environment for both validation utilities, type **setenvproc.bat** or **setenvproc.sh**. By default, the validation utilities log to a file named `icrmr.validator.log` in the WebSphere logs directory. You can modify the level of information logged and the location of the output in the `icrmr_validator_logging.xml` file. Be sure that the user ID that you use to run the utility has read permission to the XML file and write permission to whatever log file that you configure for use. The `icrmr_validator_logging.xml` file is installed with the Resource Manager code in the WebSphere Application Server `installedApps` path, subdirectory `/icrmr.ear/icrmr.war/`.

Details about how to use the utilities and understand the reports is discussed in 6.3.2, “Content Manager recovery” on page 208.

2.6.5 Tivoli Storage Manager backup

If you are using Tivoli Storage Manager to store your documents in secondary storage media, you must remember to back up the data that is residing in this component as well. We highly recommended that you read Part 5, “Protecting the Server,” in *IBM Tivoli Storage Manager Administrator's Guide*, GC32-0782, for detailed information about this topic.

In summary, there are a few different components you have to take into account when protecting your Tivoli Storage Manager servers:

- ▶ Tivoli Storage Manager database

Tivoli Storage Manager stores all the information about each stored file in an internal database. If this database is lost, all the information stored in Tivoli Storage Manager-managed devices is inaccessible. For this reason, it is very important that you back up the Tivoli Storage Manager database regularly. There are several options to do this listed in the Tivoli Storage Manager documentation mentioned earlier.

- ▶ Tivoli Storage Manager storage pools

The information stored into Tivoli Storage Manager actually resides in storage pools defined in Tivoli Storage Manager. These pools include disk storage pools, manual tapes, and automated libraries. All this information can be protected by creating a Tivoli Storage Manager copy storage pool, which will hold a copy of the information available in the original storage pools. You must make sure the information in the copy storage pools gets updated periodically using the different options available in the Tivoli Storage Manager server. Also, it is a good practice to take the copy volumes off-site to prevent them from having the same risk of failure as the original copies.

- ▶ Tivoli Storage Manager configuration and history files

Some information is not available as part of the Tivoli Storage Manager database and also must be saved in a secure place to allow the recovery of the Tivoli Storage Manager server in case of a failure. These files are:

- Tivoli Storage Manager volume history
- Tivoli Storage Manager device configuration file
- Tivoli Storage Manager server options files

For some of these files, you need to first run a Tivoli Storage Manager server command to update the file with the latest information before saving it to a backup location. These files will be needed when a Tivoli Storage Manager recovery is performed before the database backup can be issued.

- ▶ Database, log, and disk volume information

You probably want to record the information about the location, name, and size of the different volumes that existed on the Tivoli Storage Manager server hard drive to hold the Tivoli Storage Manager database, the database logs, and the different disk pools before the server crashed. Even though this information does not need to be exactly the same when you are doing a Tivoli Storage Manager recovery, it might come handy to reproduce the same or a similar environment as the original one.

2.6.6 Information Integrator for Content configuration database

Content Manager Information Integrator for Content can enhance a Content Manager solution with different services. Currently, it offers:

- ▶ Federated search capabilities
- ▶ Text mining features, including:
 - Language identification
 - Summarization
 - Clustering
 - Categorization
 - Advanced search
- ▶ Advanced workflow integration with WebSphere MQ workflow

When using any of these services, the Information Integrator for Content will create its own database, frequently called a configuration database. It is a DB2 database and can be handled as described in 2.3, “Library Server and DB2 database backup and recovery” on page 14.

When creating this configuration database on the same system as your Content Manager Library Server, you can choose if both databases should be merged into one. If you choose to store the Information Integrator for Content configuration database into your Library Server database, you will create a backup of both components by backing up only a single database. Otherwise, create a database backup of the Information Integrator for Content configuration database, too.

In addition to this database, Information Integrator for Content uses several configuration files. All are stored in the %CMCOMMON% directory and will also be backed up by the scripts provided in Chapter 3, “Practical backup and recovery procedures” on page 61. These files are shown in Table 2-8.

Table 2-8 Information Integrator for Content configuration files

File name	Location	Component	Purpose
cmbcmenv.properties	%CMCOMMON%	LS and I14C	Points to other config files.
cmbfedenv.ini	%CMCOMMON%	I14C	Contains connect user IDs and passwords.
cmbds.ini	%CMCOMMON%	I14C	Contains list of available configuration databases.
cmbcs.ini	%CMCOMMON%	I14C	Usage of RMI.

2.6.7 Content Manager eClient

If you are using the Content Manager eClient, the Web-based client to IBM content management solutions, you might want to back up this component as well. The eClient is basically a set of Java beans, JSP pages, and a few configuration files. Because the Java beans and JSP pages are typically static, you can reinstall the eClient code if you experience problems. If you have modified any part, you should create a backup of these files.

The eClient also uses two configuration files, which are located in the eClient installation directory. These two files are `IDM.properties`, containing server definitions and feature settings, and `IDMadmindefaults.properties`, containing the settings of which viewer to use based on the document MIME type.

The configuration of the eClient within WebSphere Application Server can be saved by the WebSphere command **backupConfig**. This command can be found in the WebSphere bin directory and is also included in our backup scripts shown in Chapter 3, “Practical backup and recovery procedures” on page 61.

2.6.8 User applications and custom code

Because user applications and custom code are very diverse, we cannot provide any recommendations about how to back them up. This is just a reminder that you also need to back up any custom-developed applications.

Practical backup and recovery procedures

This chapter provides the procedures for backup and recovery of a Content Manager system from a practical point of view.

Using the concepts introduced in Chapter 2, “Backup and recovery strategies and options” on page 7, we describe the procedures to back up each of the components of an IBM content management system. In addition, we describe the steps and considerations from the higher-level point of view of an entire system with all its components.

The procedures described in this chapter include:

- ▶ Backup and recovery procedures for:
 - Content Manager Library Server
 - Content Manager Resource Manager
- ▶ System-wide steps for backing up and recovering:
 - One-box Content Manager systems
 - Multibox Content Manager systems
- ▶ Different backup strategies for each scenario and component

3.1 Component overview

This section explains component-level backup procedures for the Library Server, the Resource Manager, a Tivoli Storage Manager server, and other possible options. It provides you with the guidance for backing up and restoring one piece of Content Manager. Because Content Manager is a complex product, involving several components that needs to be synchronized, component-level procedures are not always appropriate. In 3.5, “Multiple node configurations” on page 86, we show you how to create a consistent backup across the different components, in single system and multiple box configurations. Figure 3-1 shows an overview of all components we discuss in this chapter.

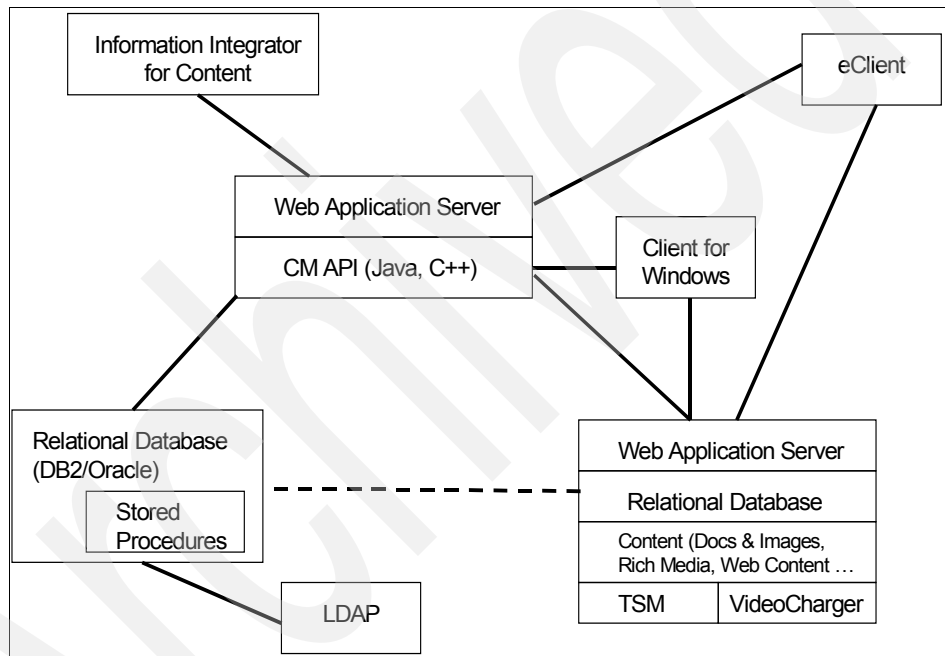


Figure 3-1 Content Manager components

3.1.1 Library Server

The Library Server is the central source for indexing, describing, locating, organizing, and managing enterprise content. A DB2 database catalogs all information and information about content stored into the system. It supports the Content Manager data architecture, controls access to content, and manages transactions.

Losing a Library Server database means losing all the content that is stored into the system. Without the Library Server database, it is impossible to access any objects on the Resource Manager. Backing up the Library Server database is one of most important backup procedures.

The Library Server itself consists of several different components. Figure 3-2 shows components belonging to the Library Server.

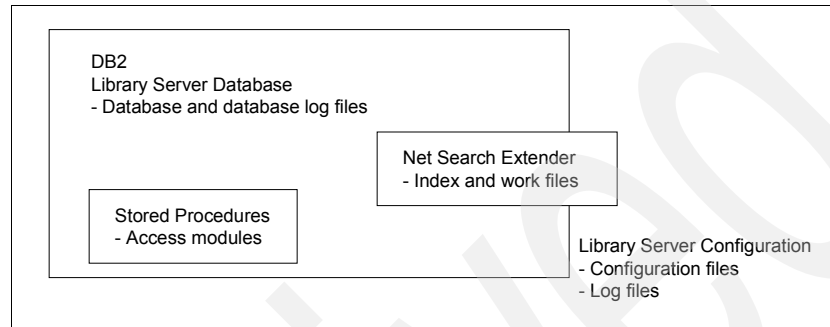


Figure 3-2 Library Server components

The Library Server components include:

- ▶ *Library Server database and database log files*: This is the database used to store all indexing, describing, locating, and organizing information. It is a DB2 database and should be backed up as a standard DB2 database. Depending on your requirements, the database log files might also need to be backed up.
- ▶ *Content Manager access modules*: The access modules are generated whenever an item type is being created or updated. They implement pre-evaluated access information to the data in the item type. Although it is not mandatory to back up these modules, it reduces the time needed to restore the system. Access modules are files generated on the Library Server machine and can be backed up as a standard file.
- ▶ *Net Search Extender (NSE) index and work files*: If using the full text indexing capabilities of Content Manager, NSE will create index and work files for every index created. These files can be backed up as any other file, but you must ensure these files are not updated during the backup window. NSE will not be able to resume operation with inconsistent files. If you lose an index, you can drop and recreate the index manually and have every object reindexed. This is a very time-consuming and resource-consuming process and is not recommended.

- ▶ *Library Server configuration files*: These files are typically generated when you install Content Manager and are only rarely updated later. But because outdated configuration files can prevent the server from operating, we recommend that you back up these files as well.
- ▶ *Library Server log files*: The Library Server generates log files during its operation. They contain information about startup and shutdown operations, license violations and also encountered error conditions. The log files are a valuable source to find out why a server was failing. They are not mandatory to restore the server itself.

3.1.2 Resource Manager

Resource Managers are specialized repositories optimized to manage the storage, retrieval, and archival of enterprise content. Documents, images and multimedia content are represented as resources stored in Resource Managers.

Figure 3-3 shows the following components of the Resource Manager:

- ▶ HTTP server
- ▶ WebSphere Application Server
- ▶ Tivoli Storage Manager (optional)
- ▶ VideoCharger (optional)
- ▶ Resource Manager Web application
- ▶ Storage and staging volumes
- ▶ Resource Manager database

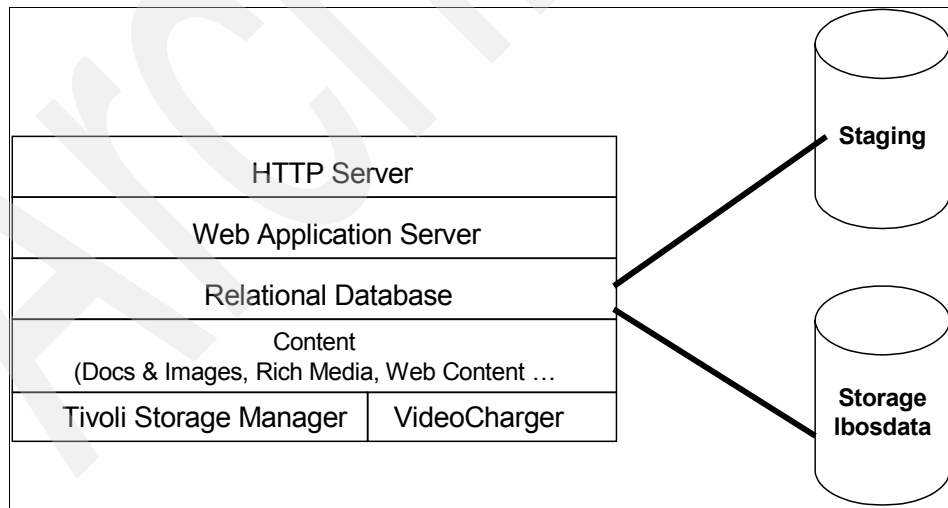


Figure 3-3 Resource Manager structure

Although there are many components involved, we focus on how to back up the Resource Manager database, storage areas, and essential configuration files. The key components are:

- ▶ *Resource Manager Web application*: Because it is a static application and it can be easily reinstalled, we focus only on its configuration data. We need to take care of the icrmr.properties file, which contains the login information, as well as pointers into the Tivoli Storage Manager environment.
- ▶ *Resource Manager services*: These are also non-changing applications. They are configured by XML files (for logging and tracing purposes) and by the setprocenv script, providing the runtime environment.
- ▶ *Resource Manager database*: A DB2 database used to maintain the status of the objects. It can be backed up as any other DB2 database.
- ▶ *Storage areas*: Objects stored into Content Manager are typically stored here, prior to migration to a secondary storage location if any. It is a file system that is split into several directories. A storage area can be backed up just as any other file system.
- ▶ *WebSphere Application Server*: Runs the Resource Manager Web application and has many configuration settings. WebSphere Application Server delivers a tool to export its configuration and to reimport them again.
- ▶ *HTTP server*: Sets up communication from the Content Manager client and APIs to the Resource Manager. During installation, it is configured for WebSphere and later for SSL. Backing up the HTTP server configuration increases recovery speed, but is not essential.

3.1.3 Summary of components

The following list includes all of the components we plan to back up in the following procedures:

- ▶ Library Server database
- ▶ Library Server access modules
- ▶ Resource Manager database
- ▶ Storage area
- ▶ NSE index and work files
- ▶ Content Manager configuration files
- ▶ WebSphere configuration files

For online backups, we also need to take care of the database log files.

3.2 Planning for the backup

When Content Manager creates the Library Server database, it creates a set of tables, views, and binds stored procedures to this database. By default, Content Manager places the database and the database log files into the same physical disk. This is neither recommended for performance, nor for database recovery. If this single disk fails, a database recovery to the latest point in time becomes impossible. We strongly recommend that you place database files and database log files on separate physical hard disks. To place the database log files in a different file system, run:

```
db2 update database configuration for <Library Server database name> using  
NEWLOGPATH "<new log path>"
```

Over time, a Library Server database will be updated many times. Every time new documents are added, updated, or deleted, the database distribution over files changes. This leads to openings in the database files, needing additional space, and consuming time for processing. We recommend that you periodically reorganize the Library Server database. The interval strongly depends on the usage of Content Manager, so there is no general recommendation except to do it whenever the performance decreases dramatically.

Note: For more information about performance considerations, see the IBM Redbook *Performance Tuning for Content Manager*, SG24-6949, or the IBM white paper *IBM Content Manager V8.2 Performance Tuning Guide*. To review the white paper, go to the following URL, select **Learn**, and enter the search string performance tuning guide:

<http://www.ibm.com/software/data/cm/cmgr/mp/support.html>

To reorganize a database, follow these steps:

1. Connect to the Library Server database.
2. Run the DB2 **reorg** command on every Library Server table.
3. Update DB2 statistics on every Library Server table.
4. Rebind the Library Server database.

A simple script doing this operation is shown in Example 3-1 on page 67. You need to replace the place holders “...” with lines for every additional table in the Library Server database.

Example 3-1 Reorg and rebind a DB2 database

```
db2 "connect to icm44 user icmadmin using password"
db2 "reorg table ICMADMIN.ICMSTACCESSCODES"
db2 "reorg table ICMADMIN.ICMSTACCESSLISTS"
db2 "reorg table ICMADMIN.ICMSTACTIONLISTCD"
...
db2 "runstats on table ICMADMIN.ICMSTACCESSCODES with distribution and detailed
indexes all"
db2 "runstats on table ICMADMIN.ICMSTACCESSLISTS with distribution and detailed
indexes all"
db2 "runstats on table ICMADMIN.ICMSTACTIONLISTCD with distribution and
detailed indexes all"
...
db2rbind icm44 -l log.txt all -u icmadmin -p password
db2 "connect reset"
```

The Net Search Extender, used for full text indexing of content and metadata, uses DB2 tables to store log information. Depending on the update frequency of the indexes and content, these tables can fill up quickly and needing more and more space in the database. If you do not encounter any problems with NSE, you should clean up these tables from time to time. The NSE messages are written to tables in the DB2EXT schema, named TEVENTIXnnnnnn, where n is any digit. A sample script is provided in Example 3-2.

Example 3-2 Delete NSE event tables

```
db2 "connect to ICMNLSDB user icmadmin using password"
db2 "delete from DB2EXT.TEVENTIX040109"
db2 "delete from DB2EXT.TEVENTIX060009"
db2 "delete from DB2EXT.TEVENTIX124308"
db2 "connect reset"
```

Notice that the name of the tables will be different in your installation. There will be one table for every NSE index being defined in the database.

Important: Whenever you create a backup of your Library Server, always try to back up all components at the same point in time. If you are creating an online backup, use the smallest possible time window to create a backup of all the components. Avoid unnecessary changes during a backup.

The following sections show how to create a backup of the Library Server. We start with the simplest option, backing up all components offline to a file system. In 3.3.3, "Offline backup to Tivoli Storage Manager" on page 72, we extend this scenario by using Tivoli Storage Manager. Finally, we switch to an online backup with Tivoli Storage Manager, as shown in 3.4, "Online backup" on page 82.

3.3 Offline backup

This section includes instructions about how to prepare for an offline backup, how to create it, and how to resume the components. It also shows how to restore the backed-up components.

3.3.1 Preparing an offline backup

Before being able to create an offline backup, it is necessary to stop all activities on the Content Manager system. This includes the following components:

- ▶ Library Server monitor
- ▶ DB2 Net Search Extender
- ▶ Stager service
- ▶ Replicator service
- ▶ Purger service
- ▶ Migrator service
- ▶ Resource Manager application server
- ▶ eClient server
- ▶ Content Manager Client for Windows
- ▶ Content Manager System Administration Client
- ▶ Any custom applications connected to Content Manager

Note: DB2 itself must not be stopped!

To prevent certain components from failing, we recommend the following order to shut down your system:

1. Content Manager client applications, including eClient, Client for Windows, and custom applications
2. Resource Manager services, including stager, purger, migrator, and replicator
3. DB2 Net Search Extender
4. Resource Manager application server
5. Library Server monitor

The script in Example 3-3 on page 69 shows how to stop these components on a Windows system, assuming everything is installed on the same machine and using standard names. At the end, it ensures that all the applications are disconnected from the DB2 instance by forcing all remaining connections. Run this example from a DB2 command line.

Example 3-3 Stopping Content Manager on Windows

```
net stop "IBM WebSphere Application Server V5 - eClient_Server"
net stop "ICM Migrator (RMDB)"
net stop "ICM Purger (RMDB)"
net stop "ICM Replicator (RMDB)"
net stop "ICM Stager (RMDB)"
db2text stop
net stop "IBM WebSphere Application Server V5 - icmr"
net stop "ICM LS Monitor ICMNLSDB"
db2 "force application all"
```

Example 3-4 shows similar steps for UNIX platforms. We tested this script on AIX only, so it might need modifications to run on Sun Solaris or Linux. To provide a reliable shutdown of Content Manager, the scripts can be enhanced by doing return code validation.

Example 3-4 Stopping Content Manager on UNIX (tested on AIX)

```
. ~/db2inst1/sqllib/db2profile
/usr/WebSphere/AppServer/bin/stopServer.sh eClient_Server
/etc/rc.cmrproc -act stop
db2text stop
/usr/WebSphere/AppServer/bin/stopServer.sh icmr
/etc/rc.cmlsproc -shutdown
db2 "force application all"
```

Important: Before running these two stop scripts, shut down any custom application or connected Content Manager Clients for Windows or Content Manager System Administration Clients.

3.3.2 Offline backup to a file system

In this section, we provide scripts to back up a Content Manager system, including the components listed in 3.1.3, "Summary of components" on page 65. This scripts have the following assumptions:

- ▶ All components are installed on same machine.
- ▶ All have standard location and names.
- ▶ No additional definitions such as secondary storage area existed.

If this does not match your environment, customization should be easy.

Example 3-5 on page 70 shows how to back up a system on Windows. In the first three lines, you can define the backup location and the names of the Library Server and Resource Manager databases.

Example 3-5 Offline backup on Windows

```
set Location=c:\mybackup\  
set LSDBNAME=ICMNLSDDB  
set RMDBNAME=RMDB  
  
mkdir %Location%  
  
REM Backing up the database  
db2 backup database %LSDBNAME% to %Location% >> %Location%backup.log  
db2 backup database %RMDBNAME% to %Location% >> %Location%backup.log  
  
REM Backing up the access mdoules  
xcopy "%ICMROOT%\%LSDBNAME%\*" %Location% /s /y >> %Location%backup.log  
  
REM Backing up the storage area  
xcopy "c:\lbosdata\*" %Location%\lbosdata\ /s /y /k >> %Location%backup.log  
  
REM Backing up NSE indexes  
xcopy "%DB2HOME%\db2ext\indexes\*" %Location%\indexes\ /s /y >>  
%Location%backup.log  
  
REM Content Manager configuration files  
xcopy "%CMCOMMON%" %Location%\config\ /y >> %Location%backup.log  
  
REM Resource Manager configuration files  
xcopy "c:\Program Files\WebSphere\AppServer\InstalledApps\%COMPUTERNAME%\  
icrmr.ear\icrmr.war\WEB-INF\classes\com\ibm\mm\icrmr\icrmr.properties"  
%Location% /y >> %Location%backup.log  
  
xcopy "c:\Program Files\WebSphere\AppServer\InstalledApps\%COMPUTERNAME%\  
icrmr.ear\icrmr.war\icrmr_*.xml" %Location% /y >> %Location%backup.log  
  
xcopy "c:\Program Files\IBM\HTTPServer\conf\httpd.conf" %Location% /y >>  
%Location%backup.log  
  
xcopy "%ICMROOT%\config\setprocenv.bat" %Location% /y >> %Location%backup.log  
  
REM WebSphere configuration  
"C:\Program Files\WebSphere\AppServer\bin\backupconfig" %Location%\wasconfig.zip  
>> %Location%backup.log
```

Note, this backup script does not copy any Tivoli Storage Manager-related files. It also does not backup the SSL key database for the HTTP server. During the backup, it captures output data from the command to the file backup.log, shown in Example 3-6 on page 71 as an abbreviated version. From this file, you can read the time stamps of the backups. This script overwrites previous backups in the same location, with the exception of the database backups themselves.

Example 3-6 Captured output during the backup: Abbreviated version

Backup successful. The timestamp for this backup image is : 20031121230442

Backup successful. The timestamp for this backup image is : 20031121230610

C:\Program Files\IBM\CM82\ICMNLSD\DLL\ICM05K00.bnd

C:\Program Files\IBM\CM82\ICMNLSD\DLL\ICM05K00.DLL

...

189 File(s) copied

C:\bosdata\00001\04\L1.A1001001A03H14A91449D79531.V1

C:\bosdata\00001\12\L1.A1001001A03G04B90557H93533.V1

...

2023 File(s) copied

...

NSE indexes

CM Configuration

RM Configuration

HTTP Configuration

...

ADMU0116I: Tool information is being logged in file C:\Program
Files\WebSphere\AppServer\logs\backupConfig.log

ADMU5001I: Backing up config directory C:\Program
Files\WebSphere\AppServer\config to file C:\mybackup\wasconfig.zip

ADMU0505I: Servers found in configuration:

ADMU0506I: Server name: eClient_Server

ADMU0506I: Server name: icmrm

ADMU0506I: Server name: server1

ADMU2010I: Stopping all server processes for node cm82vm

ADMU0512I: Server eClient_Server cannot be reached. It appears to be stopped.

ADMU0512I: Server icmrm cannot be reached. It appears to be stopped.

ADMU0512I: Server server1 cannot be reached. It appears to be stopped.

.....

.....

ADMU5002I: 475 files successfully backed up

Example 3-7 on page 72 shows similar steps to create a backup on UNIX systems. File locations are different from Windows, but the main procedures are the same. In this example, only the database backup time stamps are logged to the backup.log file.

Example 3-7 Offline backup on UNIX (tested on AIX)

```
. ~db2inst1/sqllib/db2profile

export Location=/mybackup
export LSDBNAME=ICMNLSDDB
export RMDBNAME=RMDB

mkdir -p $Location
chmod 777 $Location

#Backing up the database
db2 backup database $LSDBNAME to $Location >> $Location/backup.log
db2 backup database $RMDBNAME to $Location >> $Location/backup.log

#Backing up the access modules
cp -R ~db2fenc1/$LSDBNAME $Location

#Backing up the storage area
cp -R /storage $Location

#Backing up NSE indexes
cp -R ~db2inst1/sqllib/db2ext $Location

#Content Manager configuration files
cp -R $CMCOMMON $Location

#Resource Manager configuration files
cp /usr/WebSphere/AppServer/installedApps/$(hostname)/icrmr.ear/icrmr.war/
WEB-INF/classes/com/ibm/mm/icrmr/ICMRM.properties $Location
cp /usr/WebSphere/AppServer/installedApps/$(hostname)/icrmr.ear/icrmr.war/
icrmr*.xml $Location
cp /usr/IBMHttpServer/conf/httpd.conf $Location
cp /usr/lpp/icm/config/setprocenv.sh $Location

#WebSphere configuration
/usr/WebSphere/AppServer/bin/backupConfig.sh $Location/websphereconf.zip
```

You must run this script as root in order to be able to access all the different components. Note, this script does not back up any Tivoli Storage Manager-related information or the HTTP servers SSL key database files.

3.3.3 Offline backup to Tivoli Storage Manager

Following is an example procedure about how to do an offline backup of your entire Content Manager system using Tivoli Storage Manager as the backup system. It includes the necessary pre-configuration steps, some sample backup scripts and some additional recommendations.

In this example, we want to have enough backup copies to be able to restore the system up to one month back in time. We schedule the backup to run every evening and use DB2 archived logging to be able to recover to the latest point in time in case of a database only failure. Tivoli Storage Manager is used for the database backup images, archived logs, and all the files in each Content Manager component.

Note: In a simpler scenario, you might decide to use standard circular logging instead. In that case, you will still be able to recover to any of the scheduled points of offline system backup. We specify which steps of our example can be omitted in that case.

Tivoli Storage Manager server definitions

We need to set the necessary definitions in the Tivoli Storage Manager server to be able to store and maintain the backup images from the different Content Manager components. These definitions include the registration of client nodes and creation of a policy domain, two management classes, and the appropriate backup and archive copy groups.

The following actions are required:

1. Define policy domain for Content Manager backups.

It is recommended that you define a separate policy domain to handle Content Manager backups. You might also use this same policy domain for Resource Manager secondary storage. In that case, make sure you use a specific management class for Resource Manager storage.

2. Create management class for DB2 backup images and archived logs.

In our scenario, we use the default management class to store file backups. For the DB2 backup images and archived logs, we highly recommend that you create a specific management class with the copy group settings tailored for the DB2-specific operations. In our scenario, we define a management class called DB2CLASS for this purpose.

3. Create or modify copy groups.

You need to make sure that there is at least a backup copy group in the management class used for file backups, a backup copy group in the one used for DB2 backups, and an archive copy group in this last management class if you are planning to use archived logging and the DB2 user exit for Tivoli Storage Manager.

In our scenario, we set the parameters as shown in Table 3-1 on page 74 for the backup copy group in the default management class that will be used for files.

Table 3-1 Tivoli Storage Manager backup copy group for STANDARD management class

Setting	Value
Versions Data Exists	NOLIMIT
Versions Data Deleted	NOLIMIT
Retain Extra Version	30
Retain Only Version	30

We specify the values shown in Table 3-2 for the backup copy group for DB2CLASS.

Table 3-2 Tivoli Storage Manager backup copy group for DB2CLASS management class

Setting	Value
Versions Data Exists	1
Versions Data Deleted	0
Retain Extra Version	0
Retain Only Version	0

In addition, we created an archive copy group in the DB2CLASS management class specifying a Retain Version value of 7. This way, we make sure that if we lose up to six backup images, we are still able to apply a backup image that is seven days old and use the archived logs to roll forward up to the point of failure or most recent file system restore available.

It is our recommendation that you use at least two or three times the scheduled time between full database backups as the retention period for archived log files.

Of course, all the copy groups need to be assigned a copy destination, specifying which volumes will hold the data sent to the Tivoli Storage Manager server. It is very convenient to use a disk pool with a low migration threshold to your permanent backup pool so that your backup process finishes quickly and the data gets migrated to the final media as soon as possible.

Note: Do not forget to activate the policy set after you finish doing all the previous configuration steps.

4. Register client nodes.

You have to register a client node for each of the servers that run any of the Content Manager components. This client node will be used to back up the

files that reside in these boxes. For the DB2 backup images and archived logs, we highly recommend that you register a different node name given that the type of client is different for Tivoli Storage Manager (for example, a Windows client versus an API client).

In our scenario, we have only one box with the entire system, so we define a client node for the server files and another for DB2 backup. For the client node that will be used for files, there is no special consideration to take into account. The client node that will be used to do DB2 backups must be able to delete backup files (BACKDELETE=YES).

Tivoli Storage Manager client installation and configuration

The Tivoli Storage Manager client needs to be installed in each of the Content Manager components participating in your system. This component might already be installed in your Resource Managers if you are using Tivoli Storage Manager as your secondary storage manager. The Resource Manager needs to have the Tivoli Storage Manager backup-archive client installed. The Library Server also needs the Tivoli Storage Manager API client. In our scenario, we had the Tivoli Storage Manager client with both subcomponents installed in our Content Manager system.

For the backup-archive client, you only need to configure it to connect to your Tivoli Storage Manager server, probably using the host name as the client node. This is just a regular Tivoli Storage Manager client setup procedure. In addition, you should define an include/exclude list that is appropriate to the Content Manager component that you are going to back up. Example 3-8 shows an example of a possible include/exclude list to be used in a server that runs a Library Server and a Resource Manager.

Example 3-8 Sample include/exclude list for Library Server/Resource Manager backup

```
*Start by excluding everything
EXCLUDE "*/...\*"

*****
* Library Server components
*****
* Access modules
Include "C:\Program Files\IBM\CM82\ICMNLSD\...\*"

* Text indexes (if using NSE)
Include "C:\Program Files\IBM\SQLLIB\db2ext\indexes\...\*"

*****
* Resource Manager components
*****
* Local disk storage directories
```

```

INCLUDE "c:\lbosdata\...\*"

* Resource Manager configuration
Include "C:\Program Files\WebSphere\AppServer\installedApps\cm82vm\icmrm.ear\
icmrm.war\*.xml"
Include "C:\Program Files\WebSphere\AppServer\installedApps\cm82vm\icmrm.ear\
icmrm.war\WEB-INF\classes\com\ibm\mm\icmrm\*.properties"

*****
* Common configuration files
*****
Include "C:\Program Files\IBM\CMgmt\*.properties"
Include "C:\Program Files\IBM\CMgmt\*.ini"

```

One thing you can do to make this configuration better is to specify a different management class for the Content Manager data other than for the Content Manager configuration files. In the example, the first three INCLUDE lines refer to Content Manager data that is critical, but probably changes often and is large in size. Thus, you will specify a shorter retention period, such as 60 days. The rest of the files, which are only configuration files, can be kept for longer periods and even for multiple versions, allowing an administrator to check an old configuration without using much space. Other components that you should consider putting in your include/exclude list are:

- ▶ Tivoli Storage Manager client options file used by the Resource Manager
- ▶ Web server configuration files
- ▶ Digital certificates and SSL configuration

To back up DB2 databases to Tivoli Storage Manager, there are a few configuration steps that need to be followed:

1. Register environment variables.

You need to add the following environment variables in your system. For Windows systems, they should be specified as system environment variables. For UNIX systems, they need to be set up in the profile of the DB2 instance owner. Following is an example of this environment variables in our Windows Content Manager system:

```

set DSMI_DIR=C:\Program Files\Tivoli\TSM\api
set DSMI_CONFIG=C:\Program Files\Tivoli\TSM\api\dsm4db2.opt
set DSMI_LOG=C:\Temp

```

The *DSMI_DIR* directory should contain the name of the directory where the API binary files are located. You can identify this directory by looking for a file named dscenu.txt.

The *DSMI_CONFIG* variable specifies the name of the client options file that will be used by the Tivoli Storage Manager API. In this case, this means the Tivoli Storage Manager client options for DB2 backups.

The *DSMI_LOG* variable points to a directory where the *dsierror.log* file will be written. You can look here for errors with the Tivoli Storage Manager API used by DB2, such as communication or password errors with the Tivoli Storage Manager server.

2. Prepare Tivoli Storage Manager client options file for DB2.

You need to create or modify the client options file that will be used for DB2 backups and optionally for DB2 archived log files. This is the file specified in the *DSMI_CONFIG* environment variable in the previous step.

We recommend that you have a different client options file for DB2 backups other than for the regular file system backups. It is, therefore, a good idea to copy the client options file generated during the backup-archive configuration with a different name and tailor that new file for the DB2 backups.

This client options file has to have the option *NODENAME* set to the name of the client node registered in Tivoli Storage Manager for DB2 backups. Also, we recommend that you use the *PASSWORDACCESS GENERATE* option.

3. Update DB2 configuration.

The database configuration variable *TSM_MGMTCLASS* needs to be updated with the correct information before any backup. You need to do this in the Library Server database, the Resource Manager database, and any other database involved in your Content Manager system environment. Following is an example of how to configure a Library Server database to use the *DB2CLASS* management class:

```
C:\> db2 update db cfg for ICMNLSDB using TSM_MGMTCLASS DB2CLASS
```

4. Store Tivoli Storage Manager password for DB2 client node.

You need to use a program provided by DB2 to store the Tivoli Storage Manager client password assigned so that it can be used by DB2 when a backup is issued.

In Windows, open a command prompt after setting the environment variables and database configuration, change to the **c:\Program Files\IBM\SQLLIB\adsm** or equivalent directory and run the **dsmapipw** executable.

In UNIX systems, log in with the DB2 instance owner user ID after setting the environment variables and database configuration, change to the **~/sql1ib/adsm** directory and run the **dsmapipw** program.

5. Verify the connection to the Tivoli Storage Manager server.

Using the **db2adut1** program provided with DB2, perform a query to the Tivoli Storage Manager server for DB2 backup images or archived log files to make sure you do not receive any errors. If you receive any errors, look at the `dsierror.log` file located in the directory specified by the `DSMI_LOG` environment variable.

DB2 user exit configuration

If you want to keep archived database log files for rollforward recovery in Tivoli Storage Manager, you need to compile and configure the appropriate DB2 user exit (you can skip this step if you are using circular logging or not storing archived log files to Tivoli Storage Manager):

1. Modify the sample user exit program.

DB2 ships with a sample user exit to store archived log files in Tivoli Storage Manager that can be easily tailored for your environment. The source file for this user exit is called `db2uext2.ctsm` and is located in the `sqllib\samples\c` (or `sqllib/samples/c`) directory under your DB2 instance directory.

Copy this file to a temporary directory and rename it `db2uext2.c`. Then, use an editor to change the values of the `DSMI_DIR`, `DSMI_CONFIG`, `DSMI_LOG`, and `MGT_CLASS` to those used for the environment variables and database configuration in the previous step. We recommend that you spend some time reading the information available as comments at the beginning of the user exit source file.

Note that you could specify different values, such as client node name, password generation, and management class, for the user exit other than for the database backups. We highly recommend that you do not do that and instead keep the same settings for the database backup and the archived log files.

2. Compile the user exit.

After the user exit is ready, compile it following the instructions in the user exit source file. Following is an example in a Windows system:

```
C:\compile> cl db2uext2.c -i c:\Progra~1\tivoli\tsm\api\include -link  
c:\Progra~1\tivoli\tsm\api\lib\tsmapi.lib
```

Note: Do not use spaces in the path names for the compile string. Instead, replace the directories with spaces with their short version. You can use `dir /x` to obtain the correct short name for a specific directory.

After this is done, copy the resulting `db2uext2.exe` to the `sqllib\bin` directory inside your DB2 instance.

3. Enable archived logging and user exit in the database.

For each of the databases in your Content Manager system, configure it to use archived logging and the user exit you just configured and compiled. To do this, open a DB2 command window in Windows systems or log in as the DB2 instance owner user ID in UNIX and issue the following two commands:

```
db2 update database configuration for ICMNLSDb using LOGRETAIN ON
db2 update database configuration for ICMNLSDb using USEREXIT ON
```

4. Back up the database for initialization.

If you just changed from circular to archived logging (LOGRETAIN ON), you need to do a backup of your database before you can begin using it again.

Performing the backup to Tivoli Storage Manager

After the configuration is ready, you can start the offline backup at any time. Example 3-9 shows the fragments of a script to do this for a Windows system, which can be scheduled to run every evening, keeping the database images in Tivoli Storage Manager for two months. The file system objects will also be kept for two months because of the backup copy groups settings specified in “Tivoli Storage Manager server definitions” on page 73. This way, we can recover the entire Content Manager system up to its state two months before.

Example 3-9 Sample backup script fragment for a Windows system

```
set LSDBNAME=ICMNLSDb
set RMDBNAME=RMDB
set LOCATION=C:\temp

cd %LOCATION%

REM Backing up the databases
db2 backup database %LSDBNAME% use TSM >> %Location%backup.log
db2 backup database %RMDBNAME% use TSM >> %Location%backup.log

REM Backing up files
"C:\Program Files\Tivoli\TSM\baclient\dsmc.exe" incr >> %Location%backup.log

REM Cleanup old DB2 backup images in TSM
db2adutl delete full nonincremental older than 60 days >> %Location%backup.log
```

The process for UNIX is very similar, as shown in Example 3-10 on page 80.

Example 3-10 Sample backup script for a UNIX system

```
#!/bin/sh

LSDBNAME=ICMNLSDDB
RMDBNAME=RMDB
LOCATION=/tmp

chdir $LOCATION

# Backing up the databases
db2 backup database $LSDBNAME use TSM >> ${LOCATION}backup.log
db2 backup database $RMDBNAME use TSM >> ${LOCATION}backup.log

# Backing up files
dsmc incr >> ${LOCATION}backup.log

# Cleanup old DB2 backup images in TSM
db2adutl delete full nonincremental older than 60 days >> ${LOCATION}backup.log
```

As explained in “Initiating backup operations” on page 44, this is one way of starting the Tivoli Storage Manager backups. Another option not covered here is to schedule the backups using the Tivoli Storage Manager central scheduler.

3.3.4 Finishing and restarting

After completing your offline backup, it is necessary to restart your Content Manager servers and client applications. Basically, you reverse the steps used to shut down your system. A typical script for Windows is shown in Example 3-11.

Example 3-11 Restarting Content Manager on Windows

```
net start "ICM LS Monitor ICMNLSDDB"
net start "IBM WebSphere Application Server V5 - icrmr"
db2text start
net start "ICM Migrator (RMDB)"
net start "ICM Purger (RMDB)"
net start "ICM Replicator (RMDB)"
net start "ICM Stager (RMDB)"
net start "IBM WebSphere Application Server V5 - eClient_Server"
```

On a UNIX system, the script looks similar to Example 3-12 on page 81. You need to run this script as root. DB2 is already up, because it was never shut down.

Example 3-12 Restarting Content Manager on UNIX (AIX version shown)

```
. ~db2inst1/sqllib/db2profile
/etc/rc.cmlsproc
/usr/WebSphere/AppServer/bin/startServer.sh icmrm
db2text start
/etc/rc.cmrmproc
/usr/WebSphere/AppServer/bin/startServer.sh eClient_Server
```

After restarting these components, you can start custom applications. All opened applications need to reconnect to Content Manager, because their connections to the server have been terminated.

3.3.5 Restoring

If you encounter any kind of problem, making your Content Manager system unusable or inconsistent, you can use your backup images. Remember, an offline backup is a snapshot at a certain point in time. It is not possible to restore the system to the latest consistent state with only the offline backup data.

To achieve a consistent state of all components again, we strongly recommend that you restore all components of the Content Manager system, including:

- ▶ Library Server database
- ▶ Library Server access modules
- ▶ Resource Manager database
- ▶ Storage area
- ▶ NSE index and work files
- ▶ Content Manager configuration files
- ▶ WebSphere configuration files

If you only want to restore certain components from this list, you can use the tools explained in the online backup section, 3.4.3, “Restore” on page 84, to re-synchronize some components. This will greatly increase the time needed to recover, but might save you some data lost in the other case.

Before starting the restore operation, make sure all Content Manager components are stopped. You can use the same scripts, Example 3-3 on page 69 and Example 3-4 on page 69, we used to prepare the offline backups.

If you are restoring to Tivoli Storage Manager, you need to make sure the appropriate Tivoli Storage Manager clients are installed and configured as specified in “Tivoli Storage Manager client installation and configuration” on page 75.

To restore your databases, use the **db2 restore database** command. Detailed options are explained in 2.3.3, “Restoring a DB2 database” on page 28. To obtain

the latest time stamp, look at the log file created by the backup script, look at the names of your backup files, as shown in Figure 2-4 on page 23 and Figure 2-5 on page 23, or if your DB2 instance is still intact, use the command **db2 list history backup all for <database name>**. If you are using Tivoli Storage Manager for your DB2 backup images and archived log files, you can use the **db2adut1** utility to query the backup images available in Tivoli Storage Manager.

To restore other components, simply copy the files backed up using the built-in copy command of your operating system, or for example, the Tivoli Storage Manager backup-archive client. Especially on UNIX systems, make sure proper access is granted to the files.

To restore the WebSphere configuration, use the WebSphere **restoreConfig** command, which can be found in the WebSphere/AppServer/bin directories on all platforms.

To start all components after restoring the data, use the same scripts we used to restart Content Manager after creating an offline backup, as shown in Example 3-11 on page 80 and Example 3-12 on page 81.

3.4 Online backup

This section includes instructions about how to prepare for an online backup, how to create it, and how to resume the components. It also shows how to restore the backed up components, reapply database log files, and synchronize Content Manager components to achieve a consistent environment.

3.4.1 Preparing

An online backup is used to keep the Content Manager servers accessible for all users, even during backup operations. Because this scenario becomes very complex, especially for the restore considerations, we define our online scenario with the following criteria:

- ▶ Users are able to do all operations, including search, retrieve, import, update, and delete.
- ▶ Content Manager server activity should be minimal during the backup window:
 - No migration
 - No replication
 - No delete operations from the storage area
 - No full text index updates

Before being able to create this kind of online backup, it is required to enable DB2 archival logging or the user exit option. Detailed steps are provided in “Archival logging” on page 19. A short summary of the steps is as follows:

1. Disconnect all applications from the Library Server and Resource Manager databases.
2. Update the database configuration, setting LOGRETAIN to ON.
3. If using the user exit, compile and configure the user exit itself, and then update the database configuration, setting USEREXIT to ON.
4. Create an initial backup; must be of type offline.
5. Restart all applications accessing the databases.

In addition, before creating the backup, it is necessary to stop some activities on the Content Manager system. This includes the following components:

- ▶ DB2 Net Search Extender index updates
- ▶ Stager service
- ▶ Replicator service
- ▶ Purger service
- ▶ Migrator service

The script in Example 3-13 shows how to stop the Resource Manager services on a Windows system, assuming the use of standard names for the Resource Manager database. To disable full text index updates during the backup window, refer to “NSE indexes updates” on page 50. This is a planning activity and cannot be accomplished by a script easily.

Example 3-13 Stopping Resource Manager services on Windows

```
net stop “ICM Migrator (RMDB)”  
net stop “ICM Purger (RMDB)”  
net stop “ICM Replicator (RMDB)”  
net stop “ICM Stager (RMDB)”
```

Example 3-14 shows similar steps for UNIX platforms.

Example 3-14 Stopping Content Manager on UNIX (tested on AIX)

```
/etc/rc.cmrproc -act stop
```

3.4.2 Backup procedure

Creating an online backup is very similar to creating an offline backup, as shown in 3.3.2, “Offline backup to a file system” on page 69. You can use the scripts

provided in this section for creating an online backup, having the following differences in mind:

- ▶ The DB2 backup command needs to be extended by the **ONLINE** option. Simply add this to the two **db2 backup database** command lines.
- ▶ If you are not using user exit, you might want to place the database log files on highly available disks, for example, using RAID-1 or RAID-5 configurations. You can also copy these log files to a backup location manually, typically with a frequent schedule. You can query the current log file details from the database configuration.
- ▶ Files can be open or updated during the backup window. In case your file system backup utility supports a retry or ignore option for open files, consider using them. In the case of Tivoli Storage Manager, set the backup copy group copy serialization option to **SHAREDYNAMIC** for this purpose.

When creating an online backup, the order of the components being backed up becomes more important. We recommend that you back up the databases first. With the database log files, you will be able to roll forward the databases to a later point in time, for example, the time stamp of your storage area backup. This minimizes the amount of data lost or inconsistency you might experience when you need to restore your system. The scripts shown in 3.3.2, “Offline backup to a file system” on page 69 also incorporate our suggested order.

3.4.3 Restore

With online backup data, you might be able to recover to a later point in time than your last backup image. This depends on which component of the Content Manager system is failing. This section discusses the different components:

- ▶ Library Server database
If this component is failing, you need to restore the latest backup image of this database and reapply the database log files. This will recover to the latest point in time, causing no lost data. To restore the database, use the **db2 restore database** command. To reapply the database log files, use the **db2 rollforward database** command, as shown in 2.3.3, “Restoring a DB2 database” on page 28, to the end of the log files.
- ▶ Library Server database log files
DB2 will stop operating immediately, causing all transactions to end and all applications to stop working. Because the database is still intact, solve the log file problem and restart the Library Server.
- ▶ Library Server database and log files
You need to restore the Library Server database from your last backup. If using user exit or having copied the database log files to a secondary

location, reapply all available log files. You need to use the validation utilities described in 2.6.4, “Validation utilities” on page 55 to synchronize the Library Server and the Resource Managers. Data updated after the last backup and not available in the log files is lost. This is an unlikely event when you store the database and database log files on separate physical hard disks.

- ▶ Resource Manager database

If this component is failing, you need to restore the latest backup image of this database and reapply the database log files. This will recover to the latest point in time, causing no data to be lost. To restore the database, use the **db2 restore database** command and to reapply the database log files, use the command **db2 rollforward database** as shown in 2.3.3, “Restoring a DB2 database” on page 28 to the end of the log files.

- ▶ Resource Manager database log files

DB2 will stop operating immediately, causing all transactions to terminate. Because the database is still intact, solve the log file problem and restart the Resource Manager. The asynchronous recovery utility will remove any terminated transactions from the Library Server as well.

- ▶ Resource Manager database and log files

You need to restore the Resource Manager database from your last backup. If using user exit or having copied the database log files to a secondary location, reapply all available log files. You need to use the validation utilities to synchronize the Library Server and the Resource Managers. Data updated after the last backup and not available in the log files is lost. This is an unlikely event when you store the database and database log files on separate physical hard disks.

- ▶ Resource Manager storage area

This is one of the most critical data areas. Because a plain file system has no rollforward capabilities, you will only be able to recover to the point of the last backup. Because of that, we strongly recommend that you place the storage area on a high availability disk subsystem whenever you have a no data loss requirement.

There are several different restore scenarios:

- Restore the Resource Manager storage area from the last backup. Use the validation utility described in 2.6.4, “Validation utilities” on page 55 to find information about missing files. You can extract the metadata from the Library Server database.
- Restore the Library Server database, the Resource Manager database, and the storage area to a consistent point in time, which is the time of the storage area backup. You can achieve this by reapplying database log files

to exactly this point in time. Data inserted or updated after that time is lost; deleted documents might reappear.

- Custom option: Restore the Library Server database, the Resource Manager database, and the storage area to a consistent point in time. This is the time of the storage area backup. You can achieve this by reapplying database log files to exactly this point in time. Run all commands against the Content Manager server again. Some products, such as batch scanning solutions for Content Manager, save their release scripts, so you can start them again.
- ▶ Content Manager application or configuration
Reinstall the code or restore from a full system backup. Then, restore the configuration files from the backup and start your servers again.
- ▶ Library Server access modules
These modules can be restored from the backup image or recreated, as shown in “Access modules” on page 52.
- ▶ Full text index files
First, restore the index and work files from the backup location. Because the index update is triggered by an update of the table, you have to use DB2 to find the corresponding table to your data and do an update on every single row affected. The Content Manager table structure is described in *Content Manager Application Programming Reference*, SC27-1348.

3.5 Multiple node configurations

Multiple node configurations are common in production Content Manager environments. Typical layouts include a Library Server and a Resource Manager on different machines or a central Library Server and several Resource Managers in different branches of an enterprise.

For backup, the procedures available are very similar to the ones already shown to you for a single box configuration. You would use the same scripts and commands as shown in the previous part to create a backup. Additional considerations are necessary to synchronize the backup operation. If using the Tivoli Storage Manager server driven option, a central backup server will control all operations. Other options include a synchronization of server time settings or the wait for a semaphore to start the backup scripts.

Recovering a multiple node environment typically involves the recovery of the failing node, using the same steps as outlined in 3.3.5, “Restoring” on page 81. If you are able to roll forward all failing components using log files, because only the database was affected by the crash, the system is consistent again. If not,

you need to synchronize the restored node with the other Content Manager servers, using the Content Manager validation utilities, as shown in 2.6.4, “Validation utilities” on page 55.

Another recovery option is to restore all nodes to a single point in time, not only the failing one. This helps to avoid consistency problems, but increases the amount of data lost. If you are able to reload any new data since the point in time of your last backup, this might be a valuable option.

3.6 Tivoli Storage Manager backup procedure

For the situation where Tivoli Storage Manager is used only as a storage device driver for Content Manager, the following procedures can be used as a guide to help the Content Manager administrator. These step-by-step procedures can be used to back up and eventually to recover the Tivoli Storage Manager installation for your Content Manager system.

For other situations where the Tivoli Storage Manager system being used by Content Manager is used also as the network backup system for the entire organization, we do not recommend using the procedures in this redbook, but instead, refer to Part 5, “Protecting the Server,” in *IBM Tivoli Storage Manager Administrator's Guide*, GC32-0782.

Because each of the following steps can be very different in a given Content Manager system setup, we do not provide further details. Nevertheless, the general layout of the procedure almost always remains the same. Our recommendation is that you understand how the overall process works and then tailor each of the steps using Tivoli Storage Manager documentation.

You need to produce a safe copy of all the components listed in 2.6.5, “Tivoli Storage Manager backup” on page 56 to be able to recover your Tivoli Storage Manager installation in the situation that an error occurs to your Content Manager system. Following is a proposed procedure you can follow to do this. We assume that you have your Tivoli Storage Manager system already configured to support Resource Manager storage and optionally Content Manager backup images. Also, all the storage devices should be already configured.

Data backup and restore

You can make copies of the data stored on storage pool volumes to other storage pool volumes. This is useful to safeguard the data on tape or optical volumes in the case one of them gets corrupted. In the context of Content Manager systems, it is also common that Tivoli Storage Manager disk storage pools are used for mid-term and long-term storage of information. It is, therefore, also important to

back up the information in these volumes to other media that can be moved off-site.

To do this, you need to use a copy storage pool. The volumes in this storage pool will only contain information copied from the other storage pool volumes. Define the copy storage pool using the `DEFINE STGPOOL` administrative command. Following is an example of creating a copy storage pool named `BACOPY` that uses a tape device assigned to a device class named `AUXDEV`:

```
DEFINE STGPOOL BACOPY AUXDEV POOLTYPE=COPY MAXSCRATCH=0
```

With this done, you can back up the information stored on a particular storage pool or storage pool volume using the `BACKUP STGPOOL` command, as shown in the following example to back up a storage pool named `DISKPOOL`:

```
BACKUP STGPOOL DISKPOOL BACOPY
```

This process works as an incremental backup, copying only new files to the copy storage pool. This data backup can be scheduled to run automatically using either the Tivoli Storage Manager scheduler or as part of an external system backup script. In the second scenario, take into account that this backup command is an administration command and should be sent to the server through the `dsmdmc` command line interface. You might want to use the `WAIT=YES` option to make sure the backup finishes before your backup script ends.

If primary storage pool volumes are lost, mark them as `DESTROYED` using the `UPDATE VOLUME` command. Make sure the primary storage pool has enough free space; add new volumes as needed and restore the content using the `RESTORE STGPOOL` command. Another option is to use the `RESTORE VOLUME` command for each of the failing volumes, which automatically marks them as `DESTROYED` and restores the information to the appropriate primary storage pool.

System backup

To back up the Tivoli Storage Manager system information, start by deciding which storage device you will use to store Tivoli Storage Manager database backups. We recommend using a tape device, not necessarily automated, to be able to take the backup images off-site easily. You need to know the device class used to identify this storage device.

You can then start the database backup using the `BACKUP DB` command. For example, the following command creates a full database backup to device class `AUXDEV`, using a previously labeled volume named `DBBACK03` to store the backup image:

```
BACKUP DB DEVCLASS=AUXDEV TYPE=FULL VOL=DBBACK03
```

For each database backup image created, Tivoli Storage Manager writes a record in the volume history file. To be able to reuse a particular volume for a new database backup (for example, after a rotation of seven tapes has ended its cycle), you need to erase obsolete records from this file. You do this using the **DELETE VOLHIST** command. The following example erases all the records about database backup images that are more than 60 days old:

```
DELETE VOLHIST TODATE=TODAY-60 TYPE=DBBACKUP
```

It is a good idea to run this process in a scheduled basis, probably along with the system backup process.

After the database backup is performed, you need to save the volume history and the device configuration file to make a potential restore process easier. To do this, use the **BACKUP VOLHIST** and **BACKUP DEVCONF** commands to copy the latest information to a file in the server file system and then copy these files to a safe place.

Finally, it is also a good idea to save the information about the database and log volume information to have an idea of their state before the failure. You can do this by saving the output of the **QUERY LOGVOLUME** and **QUERY DBVOLUME** commands to a file and then copying this file to a safe place. Following is an example about how to do this using the Tivoli Storage Manager administrative command line interface:

```
dsmadmc -id=<usr> -password=<pass> -outfile=<file> query dbvol format=detail  
dsmadmc -id=<usr> -password=<pass> -outfile=<file> query logvol format=detail
```

System recovery

To recover your Tivoli Storage Manager server from a system failure, start by installing the Tivoli Storage Manager server including all the components that were installed prior to the failure.

Initialize the server database and recovery logs using the saved information gathered before the system crash. You can do this using the **dsmservr format** utility.

Make sure the server is stopped, copy the volume history and the device configuration files to the Tivoli Storage Manager server root directory, and change their names to **volhist.out** and **devhist.out**, respectively.

Restore the Tivoli Storage Manager database using the **dsmserv restore db** server utility and start the Tivoli Storage Manager server.

We recommend auditing all the storage pool volumes that have a possibility of being damaged, such as disk volumes that were saved after a system crash, using the **AUDIT VOLUME** command. If Tivoli Storage Manager cannot access a

particular file, and there is not a copy in a copy storage pool, it is deleted from the system and is lost. In this scenario, make sure you run the Content Manager validation utilities to bring back the Content Manager system to a consistency point after the recovery.

If there is any lost storage pool volume for which there is a backup in a copy storage pool, restore the data as specified in “Data backup and restore” on page 87.

High availability strategies and options

This chapter provides the strategies and options to achieve *high availability* for a Content Manager implementation. High availability is important to mission-critical applications. Maintaining high levels of access to content within a Content Manager environment is challenging. The purpose of this chapter is to provide the best practices and architectural patterns to achieve a highly available environment for Content Manager while balancing the costs, software, hardware, and necessary skills.

4.1 Overview

Before we describe the different high availability strategies and options for Content Manager, we need to define high availability. In this section, we provide an overview for high availability, including planned versus unplanned outages, high availability and continuous availability, cost versus loss, availability matrix, levels of availability, and measuring availability.

4.1.1 Availability concept

Availability is a measure of the time that a server or process is functioning normally, as well as a measure of the time the recovery process requires after a component failure. It is the *downtime* that defines system availability. Availability requires that the topology provides some degree of redundancy in order to eliminate single points of failure (SPOF). Although vertical scalability can provide this by creating multiple processes, the physical machine then becomes a single point of failure. For this reason, a high availability topology typically involves horizontal scaling or redundancy across multiple machines.

The concept of high availability roughly equates to system and data available almost all of time, 24 hours a day, 7 days a week, and 365 days a year. We know that 100% availability is not a cost-effective reality today for the large majority of implementations, rather, it is a goal. A reality today is closer to 99.99% availability, and we strive for 99.999% availability. Our goals are to design and build systems that are highly available by compensating for both planned and unplanned outages that can be caused by single points of failure.

Many organizations need almost continuous availability of their mission-critical applications and server resources. Loss of service (sometimes called an outage) of an important application often translates directly into lost revenue or, even worse, lost customers.

Content Manager provides availability features built-in to the product itself, primarily in the Resource Manager replication service. The product features by themselves, however, are not enough to achieve high availability for most implementations. Redundancy is a key and well-understood means of achieving high availability: redundant components, redundant systems, redundant data, and even redundant people. Hardware components can fail, and software quality varies from release to release, making other techniques for high availability equally important.

4.1.2 Planned versus unplanned outages

Outages can be broadly classified into two categories. One encompasses the planned outages that take place when the operations staff takes a server offline to perform backups, upgrades, maintenance, and other scheduled events. The other type is unplanned outages that occur due to unforeseen events such as power loss, a hardware or software failure, system operator errors, security breaches, or natural disasters. As a result, downtime can be caused by both planned and unplanned events, as shown in Table 4-1. Planned events can account for as much as 90% of downtime. Unplanned events account for 10% of downtime. The most important issue is how to minimize the *unplanned downtime*, because nobody knows when the unplanned downtime occurs, and businesses require their systems be up during normal operating hours.

Table 4-1 *Planned outages versus unplanned outages*

Type	Planned ^a	Unplanned
Software	13%	30%
Hardware	8%	15%
Application	8%	27%
Network	10%	
Operation	52%	25%
Other	9%	3%

a. Source: *IBM TotalStorage Solutions for Disaster Recovery*, SG24-6547

Unplanned failures are spread across the different components that would make up a Content Manager system or systems, such as hardware, software, and applications. It is important to understand that a redundant hardware-only solution would clearly help to prevent unplanned outages, but a true highly available system must take into account all the components of an end-to-end solution. This also includes the human factor. The human error factor can be a major contributor to downtime. Although education is important, it is also important, or perhaps even more important, to design easy-to-use system management facilities with well-documented and executed policies and procedures to help minimize any potential human factors contributing to downtime.

4.1.3 High availability and continuous availability

Confusion often occurs between high availability and continuous availability. High availability is a component of continuous availability. High availability focuses on reducing the downtime for unplanned outages, and continuous availability

focuses on continuous operation that could be seen as a “never stop” set of applications. The sum of high availability and continuous operations equals continuous availability. We address strategies and options for high availability of unplanned outages within a Content Manager environment. See Figure 4-1.

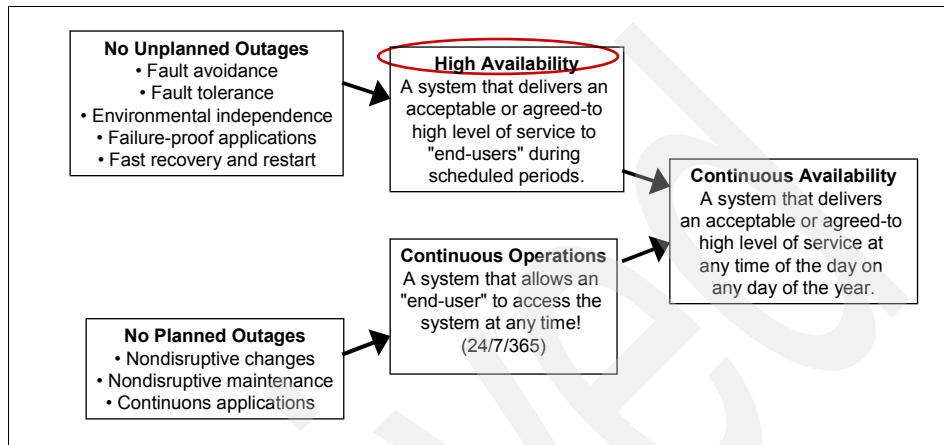


Figure 4-1 High availability and continuous availability

4.1.4 Cost versus loss

Designing high availability solutions always needs to balance the *cost versus the loss*. Although highly available systems are desirable, an optimum balance between the costs of availability and the costs of unavailability is usually required. Factoring in the intangible consequences of an outage adds to the requirement to invest in very high levels of availability. As shown Figure 4-2 on page 95, it is critical to understand the *business impact* of a failure and what the loss to the business would be if an unplanned outage did occur. Too many times as engineers we can over design a solution that might be too complex to manage and might at times contribute to an unplanned outage. Different businesses have different costs for downtime, and some businesses, such as financial services, might lose millions of dollars for each hour of downtime during business hours. Costs for the downtime include not only direct dollar losses but also reputation and customer relationships losses. Bottom line: The more you invest, the less downtime there should be to the application and business.

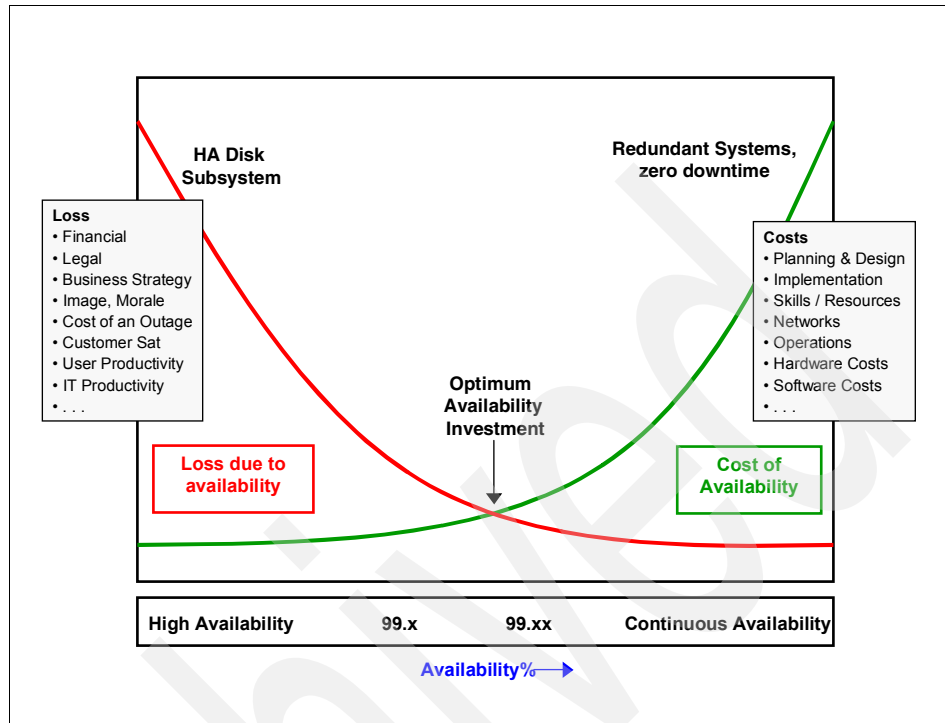


Figure 4-2 Cost versus loss

Figure 4-2 shows how this balance applies to different operating environments and when the investment in availability could reach the point of diminishing returns. At some point, the cost of availability and the loss due to availability cross to reach an optimum availability investment point. The optimum availability investment point can be a difficult equation to calculate with quantifying the losses being more challenging of the two variables. The concept, though, is to try and reach an optimum balance by designing the most cost-effective, high availability solution to support the business environment. As shown in Figure 4-2, high availability becomes increasingly expensive as you approach five or six nines and continuous availability (100%).

4.1.5 Availability matrix

We all talk about the uptime, and everybody wants 100% uptime. In reality, a 100% uptime system is prohibitively expensive to implement. For some applications, 99.9% uptime might be adequate, leaving a downtime of only 1.4 minutes per day on average or 8.8 hours per year. For some applications, 99.99% or higher uptime is required. Many people refer to 99%, 99.9%, 99.99%, and 99.999% as two nines, three nines, four nines, and five nines. The five nines

uptime is generally thought of as the most achievable system with reasonable costs, and many vendors offer such solutions. Table 4-2 shows the relationship to availability percentages and actual downtime or time loss.

Table 4-2 Availability matrix

Availability percentage	Time loss per year
99.9999% (six nines)	32 seconds
99.999% (five nines)	5 minutes
99.99%	53 minutes
99.9%	8.8 hours
99.0%	87 hours (3.6 days)
90.0%	876 hours (36 days)

4.1.6 Levels of availability

In a Content Manager environment, high availability and how to achieve high availability can mean many different things to different people. Because it is important to balance the downtime with costs, as discussed, it is also very important to evaluate what you will lose if your Content Manager service is temporarily unavailable.

Because Content Manager and content management systems in general include a variety of underlying services, subsystems, and hardware components, there are several factors and levels of high availability that can be deployed. The objective once again should be to provide *an affordable level of availability* that supports the business requirements and goals.

In a Content Manager system, there are several components that need to be taken into consideration when designing an end-to-end high availability system. For example:

- ▶ IP sprayer/load balancer
- ▶ Firewall process
- ▶ HTTP server
- ▶ Web application server and mid-tier applications (such as eClient)
- ▶ LDAP server
- ▶ Library Server application and database
- ▶ Resource Manager application and database
- ▶ Tivoli Storage Manager process and database
- ▶ Disk subsystem
- ▶ Tape and optical libraries
- ▶ Operating system processes

Content Manager high availability solutions do not use one specific technology. The solutions can incorporate a variety of strategies, most of the time within the same location, and require significant up-front planning with continual monitoring. When the solution requires servers or components to cross geographic locations, this leans more toward the disaster recovery as discussed in Chapter 6.

The technologies used must be weighed against the costs for a particular implementation to meet the business requirements. There are several levels or technologies that can be deployed today to achieve high availability for systems in general. Figure 4-3 depicts some of the more commonly used technologies with Content Manager implementations to achieve high availability.

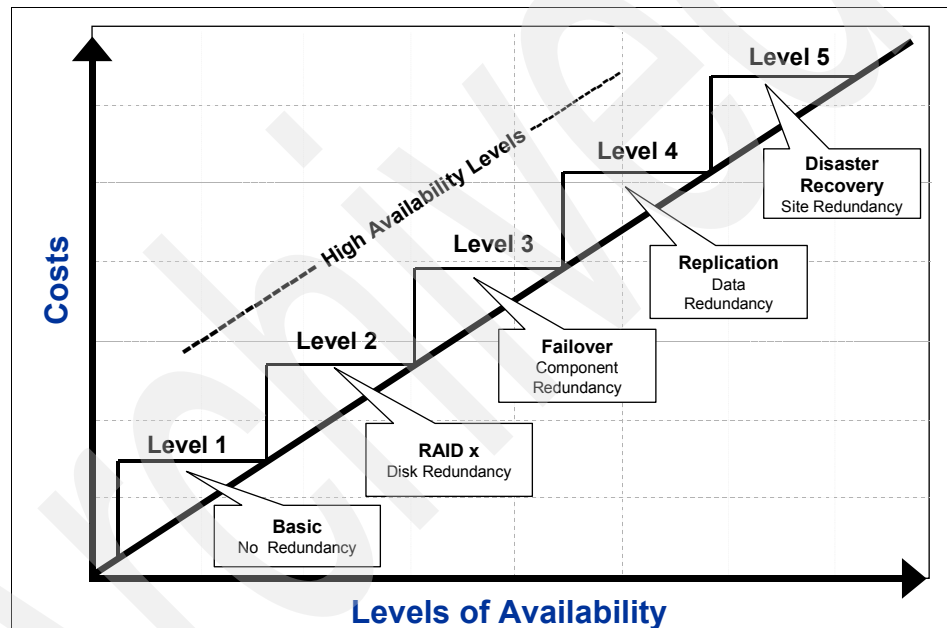


Figure 4-3 High availability tiers

We discuss levels 3 and 4 in more detail in the remaining part of this chapter in 4.2, “Content Manager HA strategies and options” on page 100, with disaster recovery (level 5) discussed in Chapter 6.

The following levels of availability are shown in Figure 4-3:

- **Level 1: Basic systems.** Basic systems do not employ any special measures to protect against data or services, although backups would most likely be taken on a regular basis. When an outage occurs, support personnel restore the system from backup (usually tape).

- ▶ *Level 2: RAID x.* Disk redundancy or disk mirroring, or both, are used to protect the data against the loss of a disk. Disk redundancy can also be extended to include newer technologies such as SANs that are emerging in the marketplace as *best practices* for high availability disk subsystems.
- ▶ *Level 3: Failover.* With Content Manager implementations, there are many components to consider that could be a single point of failure (SPOF). An outage in any single component can result in service interruption to an end user. Multiple instances or redundancy for any single component should be deployed for availability purposes. There are two primary techniques to deploy *fail-over* strategies:
 - Application clustering or non-IP cluster failover (for example, WebSphere)
 - Platform clustering or IP-based cluster failover (for example, HACMP)

Both of these are fundamental approaches to accomplishing high availability. Components that support application clustering might also be able to take advantage of load balancing in addition to availability benefits. For example, IBM WebSphere Application Server Network Dispatcher (ND) Version 5.0 and WebSphere Application Server Version 4.0 Advanced Edition (AE) has a built-in application clustering support. In order to achieve 99.9x% high availability in a Content Manager environment, you typically need to deploy both *application clustering* and *platform clustering*, assuming your solution is operating in a multi-tier or multiserver environment. If you run the full Content Manager stack on a single server (all components), platform clustering or IP-based failover would be the appropriate strategy to deploy.

In a platform clustering environment, a standby or backup system is used to take over for the primary system if the primary system fails. In principle, almost any component can become highly available by employing platform clustering techniques. With IP-based cluster failover, we can configure the systems as active/active mutual takeover or active/standby (hot spare). Additional information regarding fail-over strategies and techniques for Content Manager implementations are discussed in 4.2, “Content Manager HA strategies and options” on page 100.

- ▶ *Level 4: Replication (data redundancy).* This type of high availability for Content Manager implementations extends the protection by duplicating the database content (metadata and control tables) and file system content to another machine (server) in the event of a hardware, software, disk, or data failure. This would provide another level of protection and high availability in the event of a failure with data and content being replicated compared to a shared disk/fail-over strategy previously discussed. This type of a high availability implementation (replication) can also be used as a disaster

recovery strategy, the difference being whether the servers are located within the same location or are geographically separated.

For Content Manager implementations, this strategy would use a database replication and a Resource Manager replication strategy, as discussed later in this chapter in 4.2.5, “High availability example 3: Replication” on page 113.

- *Level 5: Disaster recovery.* This applies to maintaining systems in different sites. When the primary site becomes unavailable due to disasters, the backup site can become operational within a reasonable time. This can be done manually through regular data backups and automatically by geographical clustering, replication, or mirroring software.

It is also possible and a best practice to combine multiple high availability levels within a single solution. For example, a fail-over (*level 3*) strategy for the Library Server database and a replication strategy (*level 4*) for the Resource Manager content, with all servers using a disk redundancy strategy (*level 2*).

Examples for levels 3 and 4 are discussed in 4.2, “Content Manager HA strategies and options” on page 100.

4.1.7 Measuring availability

The total availability of an infrastructure is calculated by multiplying the availability of each component, for example, 98.5% x 98.5% x 99.5%, and so on. It is very important to balance the availability of all components in a Content Manager production system, *as perceived by the end users*. This means that true availability is the product of the components or the weakest link comprising the end-to-end solution, as shown in Figure 4-4 on page 100. Do not over spend or over engineer on any particular component, and do not under spend or under engineer on other components. For example, the system shown in Figure 4-4 on page 100 has a system availability of 87% as seen by the end user, compared to the availability of the Content Manager server or servers at 99.6%.

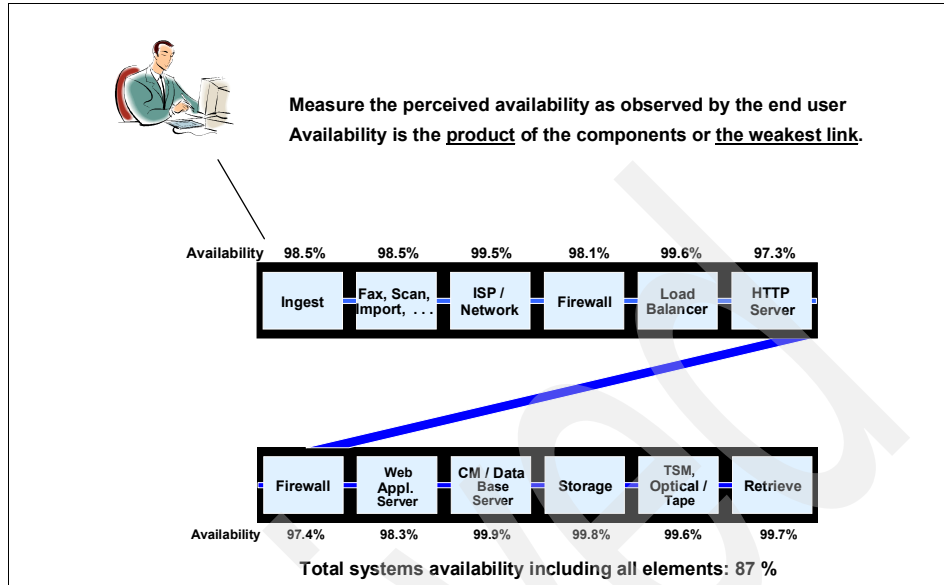


Figure 4-4 Measuring availability

Availability is not free. It takes hard work and serious management attention to integrate the many diverse components, people, and processes into a stable, highly available Content Manager system. High availability starts with reliable products. Today's Content Manager products are reliable and are capable of delivering high levels of availability, but reliable products alone will not assure high quality service. Very high levels of availability rely on an infrastructure and application design that includes availability techniques and careful system integration. A lack of, or failure to follow, careful management and effective systems management procedures is one of the most common causes of an outage. Change management, in particular, needs more emphasis. Effective management systems that employ defined and repeatable processes contribute significantly to higher levels of availability, while in the long run, actually decrease the cost of IT services through more effective and efficient use of IT resources.

4.2 Content Manager HA strategies and options

Content Manager Version 8 is built on relational database, Web application server, and Tivoli Storage Manager, as shown in Figure 4-5 on page 101. High availability for a Content Manager system includes understanding, configuring, and tuning these underlying software components, as well using the internal features of Content Manager.

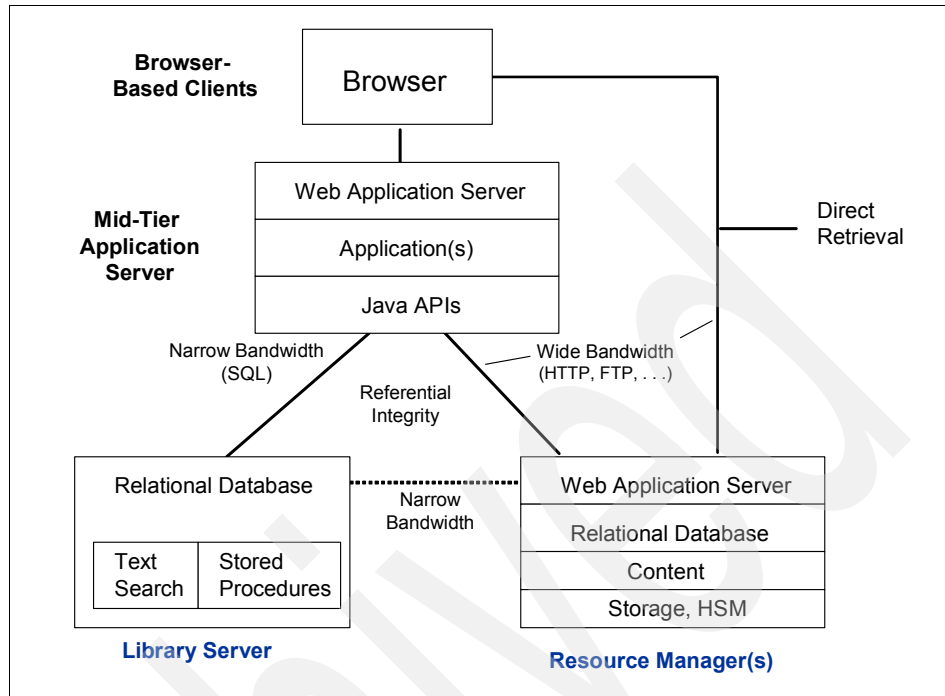


Figure 4-5 Content Manager Version 8 logical model

The role of a Content Manager architect or software specialist is to evaluate business problems and to build solutions to solve them. To do this, the architect and specialist begin by gathering input on the problem, an outline of the desired solution, keeping in mind any special considerations such as high availability and disaster recovery that need to be factored into the solution. To enable IT practitioners to do this better each time, we need to capture and reuse the experiences in such a way that future engagements can be made simpler and faster.

To this end, we provide options and strategies in this chapter using a patterns approach to depict highly available components making up an end-to-end Content Manager implementation.

4.2.1 Content Manager infrastructure fundamentals

With Content Manager Version 8, the infrastructure to support it can leverage a traditional *n-tier* architecture made up of multiple components. As shown in Figure 4-6 on page 102, Content Manager with multi-tiered infrastructure can inherently contribute to availability by segregating functions and can allow for

increased redundancy, increased scalability, and simplified management among components.

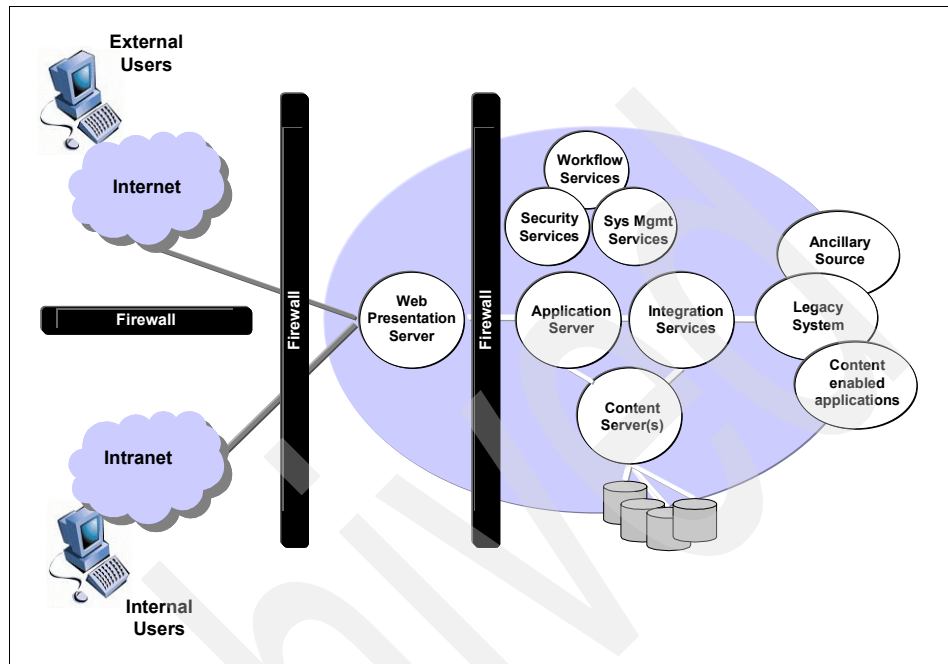


Figure 4-6 Content Manager logical tiers

There are many approaches and levels of redundancy, each with different costs and effectiveness trade-offs. As an extension, the logical model presented in Figure 4-6, a multi-tier *physical model* can be divided into three or more functional tiers, as represented in Figure 4-7 on page 103:

- ▶ Client tier
- ▶ Presentation tier
- ▶ Business logic or application tier
- ▶ Data tier

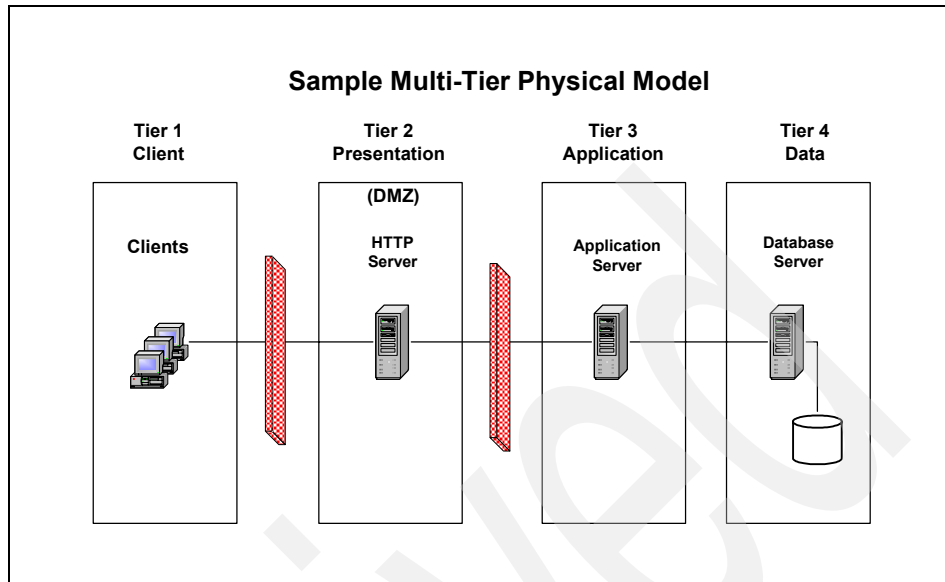


Figure 4-7 Multi-tier physical model

The *client tier* encompasses various client types, such as browsers, applets, or stand-alone application clients. These clients can reside both within and outside of the enterprise firewall. User actions are translated into server requests and the server responses are translated into a user-readable format.

The *presentation tier* provides services to enable a unified user interface. It is responsible for all presentation-related activity. In its simplest form, an HTTP server accesses data and services from other tiers, handle user requests and screen flow, and can also control user interaction. In order to separate the *presentation tier* from an *application tier*, an HTTP server is used in conjunction through plugins with a Web application server, such as WebSphere. The advantage of using a separate HTTP is that you can move the application server behind the domain firewall into the secure network, where it is more protected than within the DMZ.

The *business logic or application tier* embraces Web components, JSPs, beans, or servlets, which are deployed in Web containers. The Web application server provides the infrastructure for application and business logic. It is capable of running both presentation and business logic, but generally does not serve HTTP requests.

The *data tier* is commonly referred to as the back-end tier; examples include database manager systems, *Content Manager* systems, mainframe transaction processing, and other legacy systems.

In a Content Manager system, there are several components within the tiers that need to be taken into consideration when designing an end-to-end high availability system. For example:

- ▶ Client workstations
- ▶ DNS server
- ▶ IP sprayer/load balancer
- ▶ Firewall process
- ▶ HTTP server
- ▶ Web application server and mid-tier applications (such as eClient)
- ▶ Library Server application and database
- ▶ Resource Manager application and database
- ▶ Tivoli Storage Manager process and database
- ▶ Disk subsystem
- ▶ Tape and optical libraries
- ▶ LDAP servers

The scope of this redbook focuses on the components that are critical and required for Content Manager implementations. Several references and examples exist that discuss in detail other components not discussed in this redbook.

The following Content Manager high availability examples are based on experiences and reusable assets that provide a source to assist you to build high availability Content Manager solutions. The examples discussed are assumed to be a start in designing such solutions, because each solution will have its unique requirements. The examples given here are aimed at giving you an 80/20 rule when it comes to Content Manager high availability solutions.

4.2.2 High availability example 1: Clustering

The first high availability pattern shown in Figure 4-8 on page 106 uses WebSphere *application clustering* for the eClient or custom application in tier 3 and *platform clustering* for the tier 4 Content Manager components:

- ▶ *Application clustering*: Application or horizontal clustering exists when multiple Web application servers (a cluster) are located across multiple physical machines. This enables a single WebSphere application to span several machines yet still present a single logical image to the end-user application. The HTTP server distributes (balances) requests to the cluster of application servers. *Application clustering* also allows the overall system to service a higher application load than provided by a single sever configuration, because the n number of servers can load balance the transactions across the cluster in conjunction with the HTTP servers.

- *Platform clustering*: There are several platform-clustering software packages available primarily based on the operating system. The unit of failover usually includes a collection of network definitions and disk storage, and one or more services such as Content Manager application, database server, HTTP server, WebSphere Application Server, and operating system file systems. However, it is not standardized, and different vendors use different terms. For example, the unit of failover in the Sun cluster is called a logical host, and the unit of failover in an IBM HACMP or HACMP/ES cluster is called an application server. The concepts, however, are the same: In a platform clustering environment, a standby or backup system is used to take over for the primary system if the primary system fails.

A fail-over platform cluster runs an application on only one primary node in the cluster at a time. Other nodes might run other applications, but each application runs on only a single node. If a primary node fails, the applications running on the failed node fail over to another node and continue running. With this approach, we can configure the systems as active/active (mutual takeover) or active/standby (hot standby):

- *Active/active mode*: Two services reside in two nodes (for example, Library Server on machine A and Resource Manager on machine B) that are configured as mutual failover.
- *Active/standby mode*: One node is configured as the primary to run full Content Manager stack, and the other node is configured as a hot standby.

The configuration for both modes is very similar. The advantage of the active/active mode configuration is lower hardware costs; the disadvantage is that the service performance is reduced when a failover occurs. The advantage of the active/standby mode configuration is steady performance; the disadvantage is that redundant hardware is needed. Furthermore, the active/active mode configuration can have twice as many interruptions as the active/standby mode configuration, because a failure in any of two nodes might cause a failover.

During a Content Manager *fail-over* scenario, the application or applications processes that move from the primary system to the backup system includes several steps:

1. Stop and exit the failed process.
2. Release the resources.
3. Detach the disk array.
4. Reattach the disk array to the backup system.
5. Check the disk and file system.
6. Repair the data integrity.

7. Gain all resources for running the process in the backup system.
8. Start the process in the backup system.

This fail-over process can take several seconds to several minutes after the fault is detected.

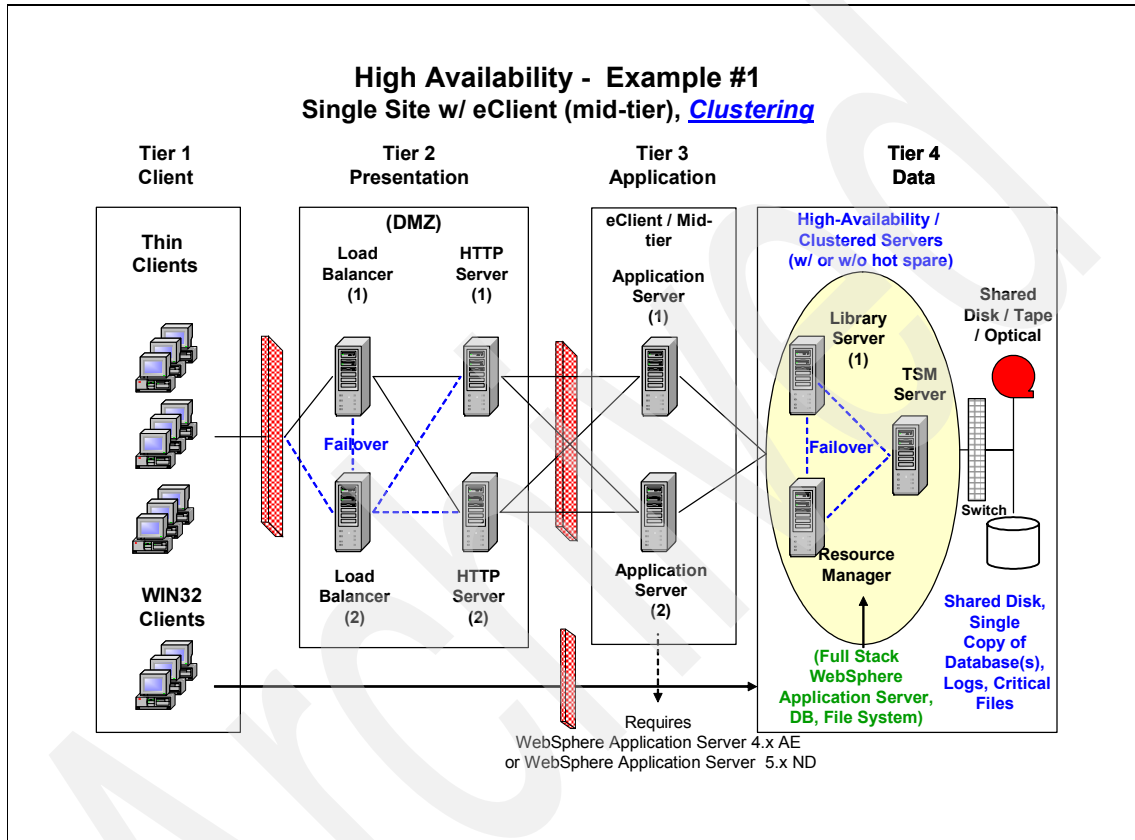


Figure 4-8 High availability: Example 1, clustering

The components and the high availability options for this example are listed in Table 4-3.

Table 4-3 Example 1: High availability component matrix

Component	High availability technique used
Domain Name System (DNS)	Uses one primary and any number of backup DNSs. A DNS supports multiple sites by passing a different IP address every time it is queried with a name (DNS round-robin selection).
Load balancer	Uses a high availability implementation with a primary and backup load balancer (IP sprayer). The backup would be a similarly configured load balancer that has a heartbeat with the primary. If the primary load balancer fails, the backup initiates a take over function. This feature is available in most load balancers and has been implemented by the WebSphere Edge Server Network Dispatcher. It is a software-based solution that provides dynamic load balancing across multiple servers, groups of servers, and geographic locations in a highly scalable, highly available fashion. As a software solution, it is highly adaptable to an enterprise's particular needs. Network Dispatcher dynamically balances not only HTTP traffic, but SSL and FTP as well.
Firewall service	Use redundant firewalls to provide load balancing and availability benefits. The protocol firewall is typically implemented as an IP router and is basically configured with filters. It protects from access to unauthorized services in the DMZ and also can avoid inadequate LAN bandwidth usage. The domain firewall prevents unauthorized access to servers on the internal network by limiting incoming requests to a tightly controlled list of trusted servers in the DMZ. In a multi-tier architecture, it prevents the user from accessing any critical data or application directly.
HTTP Web server	Implement redundant servers, load-balanced with a load balancer (IP sprayer) to give greater availability and throughput. Each server should be able to handle any request identically.
Web application server and mid-tier application (for example, eClient)	Implement multiple Web application servers, and then use a mechanism to allow servers to share the workload. The HTTP traffic must be spread among a group of servers. These servers must appear as one server to the Web client (browser), making up a cluster, which is a group of independent nodes interconnected and working together as a single system. A load balancing mechanism such as an IP spraying, as previously discussed, can be used to intercept the HTTP requests and redirect them to the appropriate machine on the cluster through an HTTP plug-in, providing scalability, load balancing, and fail over as part of the WebSphere or application server of choice. This is an example of an application cluster as discussed earlier. To support a WebSphere application cluster, you need to implement to WebSphere Version 5.0 Network Dispatcher or Version 4.0 Advanced Edition in this tier. The versions that ship with Content Manager, free of charge, Version 5.0 Base and Version 4.0 Advanced Edition Single Server (AEs) do not support application clustering.

Component	High availability technique used
Library Server application and database	In this example, tier 4 components are implemented using a platform clustering software package, such as HACMP, Veritas clustering, or Microsoft Cluster Service (MSCS), as discussed earlier. The concepts are the same. In a platform clustering environment, a standby or backup system is used to take over for the primary system if the primary system fails. For the Library Server application, this primarily means a DB2 or Oracle database fail-over implementation, in which all critical components and dynamic content need to be installed and accessible on a shared disk technology. This would include the database instance, the database itself, and database logs. The database and application binaries are recommended to be installed on the internal or operation system drives (for example, rootvg or d:\program files). This can be installed in an active/active mode, or in an active/standby mode.
Resource Manager application and database	In this example, the full stack or products making up the Resource Manager is also implemented using a platform clustering software package. The full stack meaning DB2 or Oracle database, database logs, WebSphere Application Server, Resource Manager application, and file system content (for example, ubosdata and staging). The same principles apply as with the Library Server, in which all critical components and dynamic content need to be installed and accessible on a shared disk technology. Because this example is using a platform cluster approach, the Resource Manager could use the WebSphere Application Server that comes shipped with Content Manager, Version 5.0 base or Version 4.0 AEs, compared to Version 5.0 Network Dispatcher or Version 4.0 Advanced Edition, which are required when application clustering is required.
Tivoli Storage Manager server and database	In this example, Tivoli Storage Manager is also implemented using a platform clustering software package, as shown in Figure 4-8 on page 106. Tivoli Storage Manager includes the Tivoli Storage Manager database, database logs, any disk pools, and library pools. The same principles apply in which all critical components and dynamic content need to be installed and accessible on a shared disk technology. The one additional requirement is that if optical or tape libraries are involved, a SCSI, SAN switch, or twin-tail connection will be needed in order for both the primary and backup machines to have physical connectivity in the event of a host failure.

4.2.3 Example 1 modified: Clustering with mid-tier RM application

The second example for high availability, shown in Figure 4-9 on page 110, is a modified pattern using example 1 discussed in 4.2.2, “High availability example 1: Clustering” on page 104. The difference with this pattern is the Resource Manager WebSphere *application* is moved from a tier 4 server to a tier 3 application server. In other words, this is a split Resource Manager

implementation. The Resource Manager database, database logs, file system content, Tivoli Storage Manager server, and Tivoli Storage Manager content all resides in the data tier, while the Resource Manager *application* leverages the clustering features in a WebSphere application tier. This provides the Resource Manager application with automatic load balancing, fail-over protection, and application isolation from the database tier. This also provides more consistency with a true multi-tier implementation in which applications executed in the *application tier*, as shown in Figure 4-9 on page 110.

Although there are advantages with this modified pattern, it is also important to note that the Resource Manager application servers will need concurrent access to the single Resource Manager database, file system content, and Tivoli Storage Manager content residing on the backend server or servers. Concurrent database access is a standard feature of a mid-tier application server. Concurrent access to a Tivoli Storage Manager server would also be a standard feature, but will require a Tivoli Storage Manager client residing on the n number of mid-tier servers. Finally, concurrent access to file system content will require a file system lock manager. If using a SAN solution as the disk storage subsystem, lock management functionality is provided by most SAN vendors such as the IBM ESS subsystem. If the proposed implementation does not make use of a SAN disk subsystem, you will need to ensure the mid-tier application servers have concurrent access to the shared file system content in order to leverage this example.

All of the other components, tiers, and high availability features would remain the same as discussed in example 1.

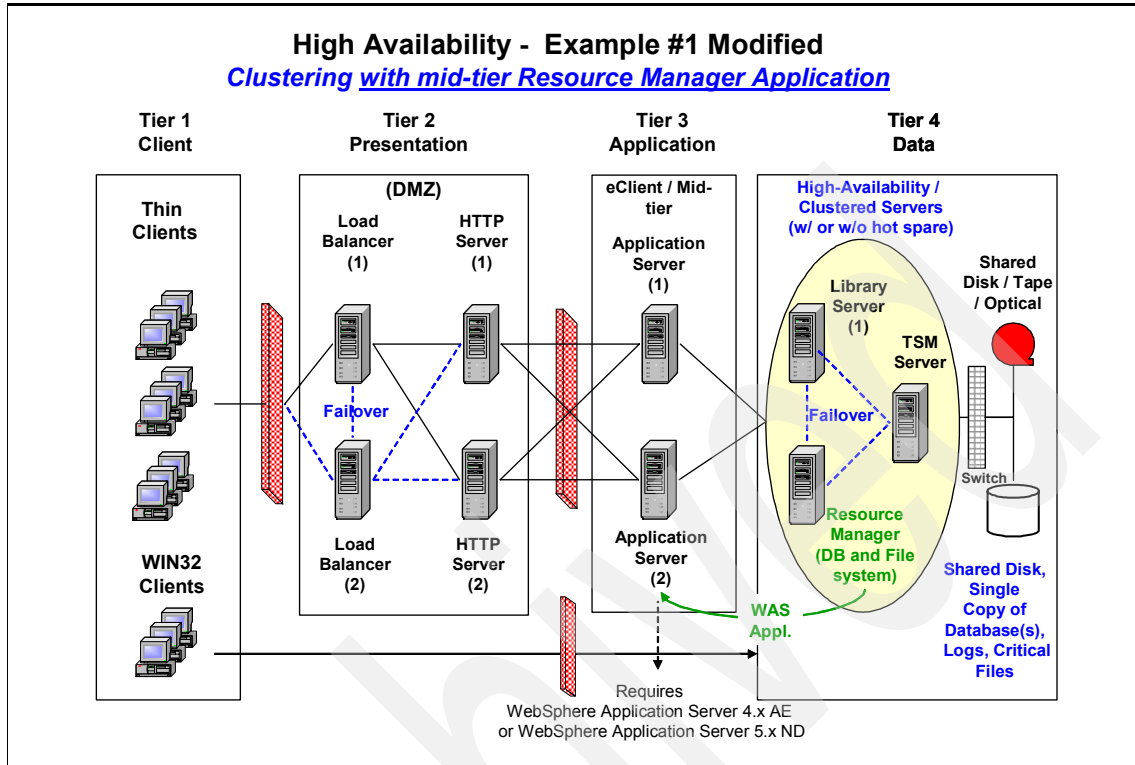


Figure 4-9 High availability: Example 1, modified platform clustering with mid-tier Resource Mgr Application

4.2.4 High availability example 2: Clustering and replication

The next pattern, example 2 shown in Figure 4-10 on page 111, uses WebSphere *application clustering* for the eClient or custom application, *platform clustering* for the Library Server component, and Resource Manager *replication* for the actual content.

In a Content Manager implementation, the content is the most vulnerable data within the solution. With database content, it is recommended to have all databases in a roll-forward recovery mode in case of a database or file system failure. This enables you to restore a backup copy of the database (DB2, Oracle, or Tivoli Storage Manager) and then apply the database logs, restoring to the point of failure or the last committed transaction, ensuring the data integrity of the system at the database level. With file system content, there are no roll-forward logs; therefore, replication of the content adds an additional level of protection in the event of disk or file system failure.

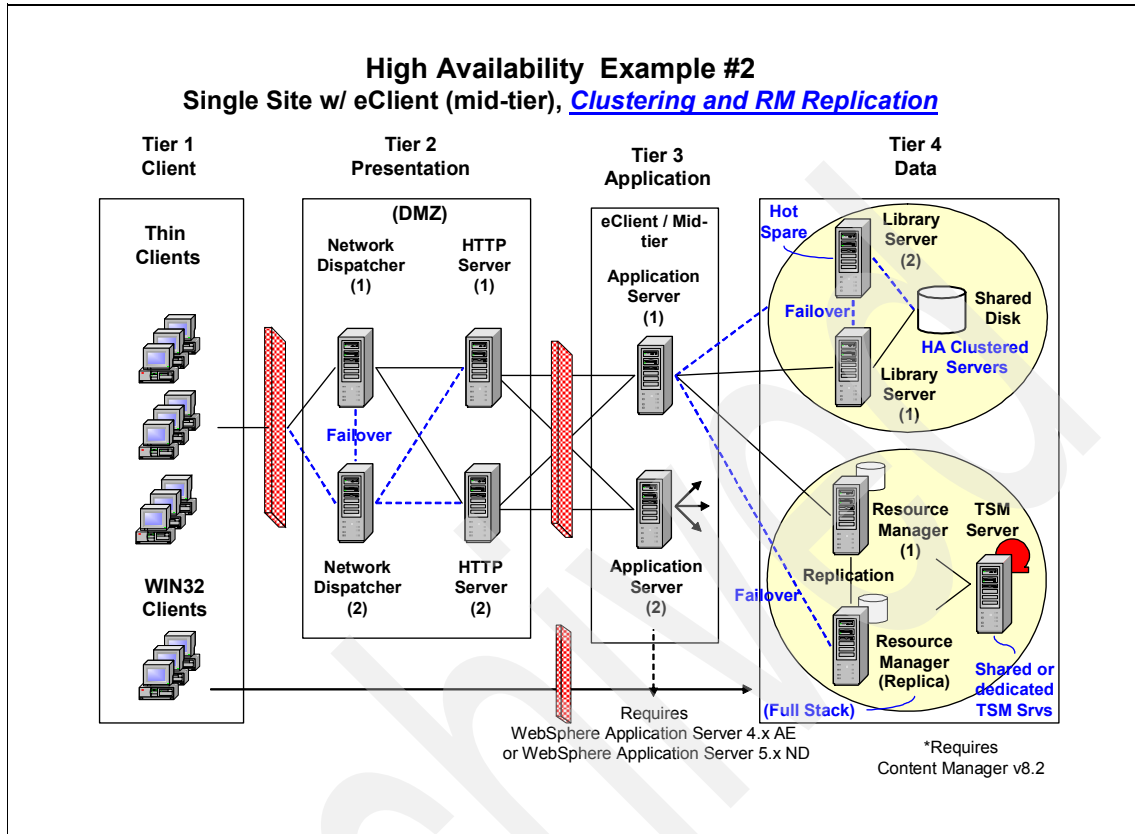


Figure 4-10 High availability: Example 2, clustering with Resource Manager replication

With Content Manager Version 8, Resource Manager replication has been enhanced over previous versions, making it a very viable high availability and content redundancy option. As shown in Figure 4-11 on page 112, content replication can be enabled to store copies of objects in multiple Resource Managers. This enhances retrieval performance and provides additional fail-over protection in case of a Resource Manager failure. In this situation, clients will automatically be directed by the Library Server to secondary Resource Manager with replicated objects.

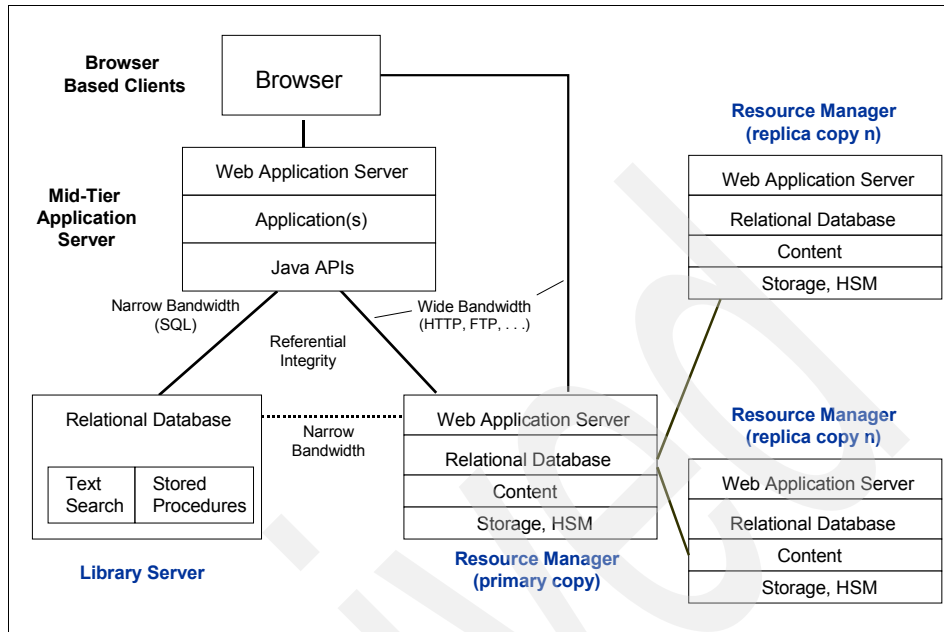


Figure 4-11 Resource Manager Version 8 replication

The Library Server has a monitoring process that will regularly check (ping) all the Resource Managers in its domain to verify their state. If it detects that a Resource Manager is down, the Library Server marks it as being unavailable in a Library Server table. All future requests for an object will be directed to the next Resource Manager in the Library Server replication table. This makes the time and action to retrieve an object transparent to the user. Also, users and applications can store content to a backup Resource Manager in the event that the primary Resource Manager is unavailable. The replication agent will automatically synchronize the objects back to the primary Resource Manager when the primary server comes back online. This allows for a two-way replication (when a primary server is down) making it once again a viable high availability option for content redundancy.

In the past and still today, many customers use Tivoli Storage Manager as another option for content redundancy, creating one or more backup copies of objects in the event of a file system failure or corruption. A Tivoli Storage Manager redundancy strategy is also a very viable strategy for content high availability and can offer advantages that Resource Manager replication is not typically used for today, for example, redundant content within a single server through Tivoli Storage Manager backup copy pools.

All of the other components, tiers, and high availability features would remain the same as discussed in example 1.

4.2.5 High availability example 3: Replication

The next pattern, example 3 shown in Figure 4-12, is similar to example 2 discussed in 4.2.4, “High availability example 2: Clustering and replication” on page 110, except it uses *database replication* for the Library Server component versus a fail-over strategy. This coupled WebSphere *application clustering* for the eClient or custom application and Resource Manager *replication* for the content delivers a level 4 high availability strategy (data redundancy), as discussed earlier in 4.1.6, “Levels of availability” on page 96.

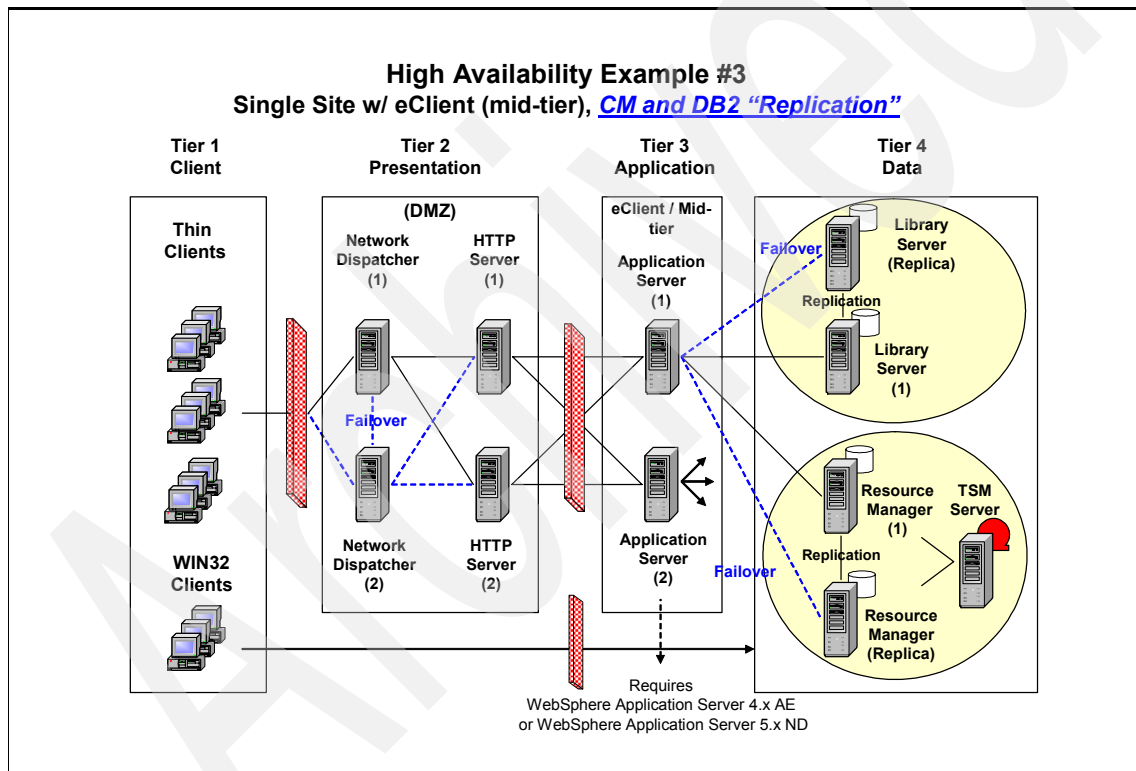


Figure 4-12 High availability: Example 3, Content Manager and DB2 replication

The difference with this example compared to example 2 is a Library Server (database) replication versus a fail-over (shared disk) implementation.

Database replication

There are several strategies and techniques that can be used for database replication. It is the intent of this section to highlight the different techniques that could be used in a Content Manager implementation. Additional information is available for DB2 and Oracle replication strategies that should be carefully researched and understood if this option is selected as the appropriate high availability strategy to pursue. There are four primary techniques that could be used to replicate the Library Server DB2 database, all with the goal of replicating or mirroring DB2 database content from a primary server to standby server:

- *Log shipping*: Log shipping is a method where transaction logs are automatically *shipped* from a primary DB2 server and made accessible on a standby server. After the log files are located on the standby server, it can stay relatively synchronized with the primary server. All database changes (inserts, updates, or deletes) are recorded in the DB2 transaction logs. Transaction logs are primarily used for crash recovery and to restore a system after a failure, but they can also play a role in replication through *log shipping*. This requires a secondary system ready to take over in the event the primary system fails. The administrator creates this secondary system by restoring a backup of the primary system database. The backup system would be placed in a rollforward pending state by restoring a database backup of the primary system. Transaction logs are then *shipped* from the primary system to the secondary system and used to roll the database forward through the transactions. In the event of a failure, the administrator stops the process of rolling forward, and the database is brought online.

In addition, the process of failover is typically not automated. After a failure, a system administrator must make a change at the application layer or DNS layer so that users can work against the secondary system, or make changes at the secondary system so that it can mimic the primary. This allows the application to continue to work as before with no necessary application coding changes.

- *Log mirroring*: DB2 has the ability to perform dual logging. This method to replicate a DB2 database is to exploit this dual logging capability of DB2. Similar to *log shipping*, but when this feature is used, DB2 writes the same log record to two locations simultaneously, thus producing a mirrored log environment and ensuring no loss of data. One of these servers or locations is typically a remotely mounted file system. This allows the database to create *mirrored* log files on different volumes or different systems, thereby increasing redundancy. This method does not create two active systems because the backup system is in an unusable state until it is brought out of rollforward pending state by the administrator similar to *log shipping*. The

downside of this approach is the performance cost associated with performing two disks writes, one of which might be remote.

Once again, the process to switch to the secondary server is typically not automated and will need a system administrator to make a change at the application or DNS layer so that users can work against the secondary server.

- *Database replication:* DB2 includes integrated replication capabilities. The DB2 implementation of replication consists of two pieces: Capture and Apply. The replication administrator designates replication sources from tables to be replicated, and then creates replication subscriptions on a target database, the secondary system, using the replication sources from the previous step as its source. The *Capture* process monitors the transaction logs for all changes to the replication source tables, placing any changes made to these tables into staging tables. The *Apply* program reads the staging tables and moves the changes to the subscription target on a timed interval.

As in the case of *log shipping* and *log mirroring*, *data replication* is an asynchronous process. For a period of time, either while the changed data has not yet been placed in the staging tables, or while the Apply program has not replicated the changes to the target system, the two databases are out of sync. This is not necessarily a long period of time or a large amount of data, but it must be considered a possibility.

Replication captures changes to the source tables; however, it does not capture changes to the system catalogs. For example, changes to table permissions will have to be performed on both systems, because replication is unable to replicate this change. In addition, the process of failover is not automated.

The process to switch to the secondary server is typically not automated and will need a system administrator to make a change at the application or DNS layer so that users can work against the secondary server.

There is some overhead in running replication. The amount of extra work depends on the amount of insert, update, and delete activity on the source tables. No extra locking is required on the base tables, because replication only analyzes the log files and not the base tables. But the population of the staging tables (change tables) and the logging of these transactions require database resources.

- *Enterprise storage disk mirroring:* Storage products such as the IBM Enterprise Storage Server® (ESS) using a protocol, PPRC, can allow real-time mirroring or asynchronous mirroring of data from one storage subsystem to another. The secondary ESS or storage subsystem can be located in the same site or at another site some distance away. Protocols such as PPRC are application independent. Because the copying function occurs at the disk subsystem level, the application has no knowledge of its existence. These protocols guarantee that the secondary copy of content,

database tables, and log files are up-to-date by ensuring that the primary copy will be written only if the primary system receives acknowledgement that the secondary copy has been written in a synchronous mode. These *enterprise storage subsystems* also provide similar guarantees in an asynchronous mode but with replication delays. You would typically find an *enterprise storage disk mirroring* strategy used more for disaster recovery purpose versus a high availability option. It is mentioned here for completeness and, in some cases, can be a viable option, but it is also the more expensive option of the four discussed here.

Table 4-4 summarizes the advantages and disadvantages of the DB2 replication options.

Table 4-4 DB2 replication options advantages and disadvantages

Method	Advantages	Disadvantages
Log shipping	<ul style="list-style-type: none"> ▶ Minimal impact to production system. ▶ Low cost. 	<ul style="list-style-type: none"> ▶ Transaction loss is a possibility (async process). ▶ Database is in unusable state until it is brought out of rollforward pending state. ▶ Standby database needs to be logically identical.
Log mirroring	<ul style="list-style-type: none"> ▶ No transaction loss. ▶ Minimal impact to production system. ▶ Low cost. 	<ul style="list-style-type: none"> ▶ Performance cost associated with performing two disks writes, one of which might be remote. ▶ Database is in unusable state until it is brought out of rollforward pending state. ▶ Standby database needs to be logically identical.
Database replication	<ul style="list-style-type: none"> ▶ Standby database is in a usable state. ▶ Instant failover (but without most recent transactions). ▶ Standby need not be physically and logically identical. ▶ Can choose to replicate only critical tables. 	<ul style="list-style-type: none"> ▶ Transaction loss is a possibility (async process). ▶ Extra cycles on production database for transaction capture. ▶ Some administrative actions not reflected on standby (for example, operations that are not logged).

Method	Advantages	Disadvantages
Disk mirroring	<ul style="list-style-type: none"> ▶ No transaction loss. ▶ All changes to the database (including administrative) are replicated. ▶ Shortest restart time. 	<ul style="list-style-type: none"> ▶ Performance impact of synchronous mirroring to a geographically remote site. ▶ High price (software, hardware, and network).

Note: It is important to note that with a database replication and a Resource Manager replication strategy as discussed here in example 3, there is the risk of an out-of-sync condition occurring between the Library Server and the Resource Manager or Managers. This is due to the fact that the two replication processes are independent from one another, and no process is in place monitoring the two at the Content Manager transaction level. There are ways, however, to protect or minimize this impact:

- ▶ If you run the replication process at night during a batch window, you can wait until both replication processes have completed before bringing the system back online for production use. The system would be unavailable during the replication process, but would ensure a synchronized system in the event of a failure requiring to move production to the standby servers. Also, with a nightly batch replication process, the environment could be exposed to a day's loss of work as the daily transactions are not replicated until the nightly batch job executes.
- ▶ For those environments that cannot afford to run a single nightly batch process, running the replication process on a scheduled or continuous basis is also possible. To ensure synchronization, we recommend that you use the Content Manager synchronous utility against the secondary environment in the event of a failure and before bringing the secondary servers online for production use.
- ▶ Database replication is only one example discussed in this chapter for high availability. Refer to options 1 or 2 or 4, using a database fail-over strategy versus a database replication strategy. This would ensure synchronization between the Library Server and Resource Manager components as there is only one copy of the database or databases.

All of the other components, tiers, and high availability features would remain the same as discussed in examples 1 and 2.

4.2.6 High availability example 4: Single server configuration

The final pattern, example 4 shown in Figure 4-13, represents a single Content Manager server for those installations that cannot justify or afford a multiserver configuration as shown in the previous examples. In this example, the full Content Manager stack, DB2, WebSphere, file system content, and Tivoli Storage Manager would all reside on a single server. The most cost effective strategy here would be to use a platform clustering strategy (failover) similar to example 1. This would include failing over the full Content Manager stack (and possibly the presentation tier) in the event of a failure to the primary machine, delivering a cost-effective, level 3 high availability strategy as discussed earlier in 4.1.6, “Levels of availability” on page 96. With this pattern as discussed in example 1, you can configure the fail-over environment as an active/active (mutual takeover) or active/standby (hot standby) configuration.

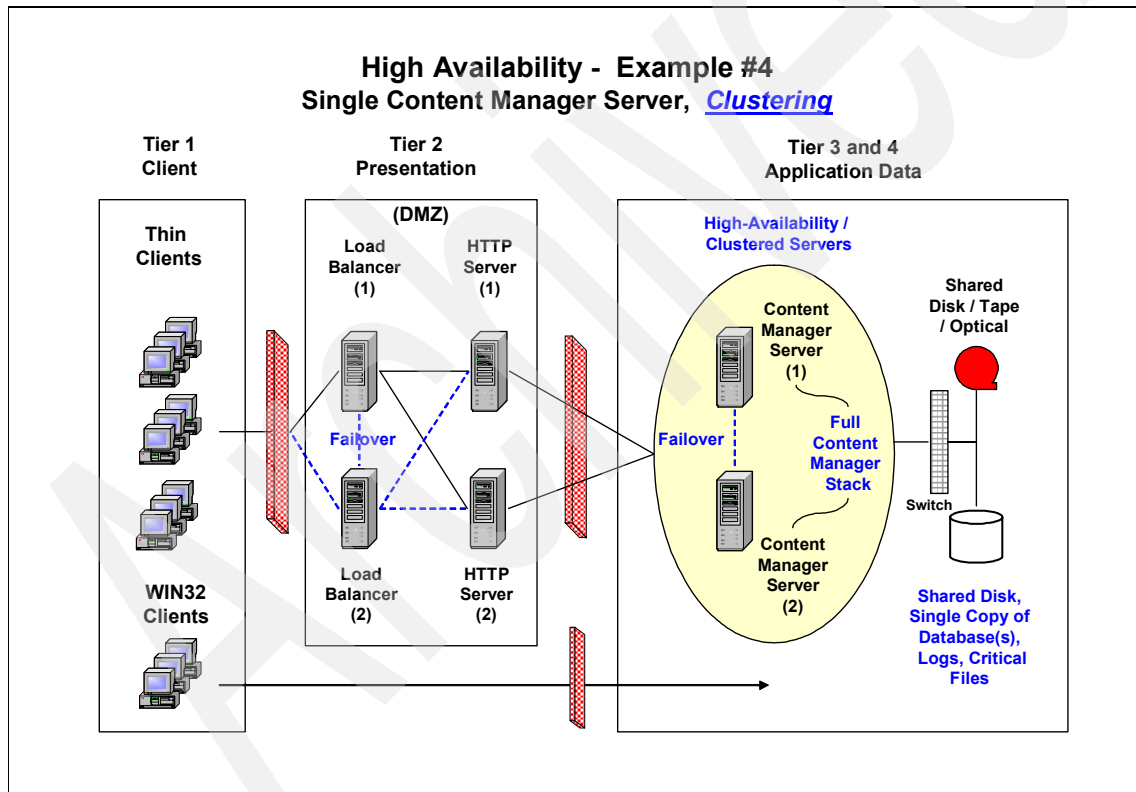


Figure 4-13 High availability: Example 4, single Content Manager server, clustering

Other high availability options exist in addition to the ones discussed here, but are either more suited for disaster recovery or are used in content management application environments (for example, Content Manager OnDemand):

- ▶ *Enterprise storage disk mirroring* (for example, SANs): As mentioned earlier in the replication options, storage products such as the IBM Enterprise Storage Server (ESS) enable synchronous or asynchronous mirroring of data from one storage subsystem to another. The secondary storage subsystem can be located in the same site or at another site some distance away. These types of solutions are application independent. The copy functions occurs at the disk subsystem level, and the application has no knowledge of its existence. Typically, these are on the more expensive end of the scale because they relate to hardware, software, networking, and skill resources to manage as compared to the other strategies discussed in this chapter.
- ▶ *High availability software mirroring strategies*: Software disk mirroring technologies, such as IBM High Availability Geographic Cluster (HAGEO), provide a flexible solution for building high availability or disaster-tolerant computing environments. Software mirroring components provide the capability for mirroring data across TCP/IP point-to-point networks over an unlimited distance from one geographic site to another. Similar to *enterprise storage disk mirroring strategies*, these are on the more expensive end of the scale because they relate to hardware, software, networking, and skill resources to manage as compared to the other strategies discussed in this chapter.
- ▶ *Dual load*: This is where you would load the data/content twice to two independent systems. This is typically a best practice strategy for Content Manager OnDemand archiving systems that impose a read-only restriction.

4.2.7 High availability summary chart

Table 4-5 shows a summary of the high availability strategies and options discussed in this chapter.

Table 4-5 High availability summary chart

HA example	Description	Level of availability ^a
Example 1	Clustering	Level 3: failover
Example 1 modified	Clustering with mid-tier Resource Manager application	Level 3: failover
Example 2	Clustering and replication	Level 3 and 4: failover and replication
Example 3	Replication	Level 4: replication

HA example	Description	Level of availability ^a
Example 4	Single server clustering	Level 3 failover: single server full stack Content Manager server
Example 5	Disaster recovery ^b	Level 5: disaster recovery

- a. Refer to Figure 4-3 on page 97 for a description of the levels of availability.
b. Refer to Chapter 6, "Business continuity and disaster recovery strategies" on page 195 for more details.

Practical procedures for high availability

In this chapter, we introduce one of the possible Content Manager high availability configurations detailed in Chapter 4, “High availability strategies and options” on page 91. We explain the steps that have to be done to set this environment up and give the results we obtained from doing different fail-over tests.

We include the configuration steps and recommendations for the following components:

- ▶ Library Server HACMP configuration
- ▶ Resource Manager replication for high availability:
 - Storage in local disk
 - Storage in a remote Tivoli Storage Manager server
 - Failover and fallback
- ▶ Different Content Manager clients:
 - Content Manager eClient
 - Windows client
 - Custom API clients

5.1 Introduction

The main goal of this chapter is to provide detailed configuration steps for implementing a highly available Content Manager system. The steps provided here can be taken by someone who is configuring any such system and who can use only the applicable steps as building blocks in that solution.

With these purpose in mind, we decide to implement and test the scenario shown in Figure 5-1, which provides the details about the configuration of at least two different options for the pieces of a Content Manager high availability setup. This scenario focuses only on the high availability of the Content Manager server components and leaves the Tivoli Storage Manager server, the mid-tier application servers, and the client components out of the scope.

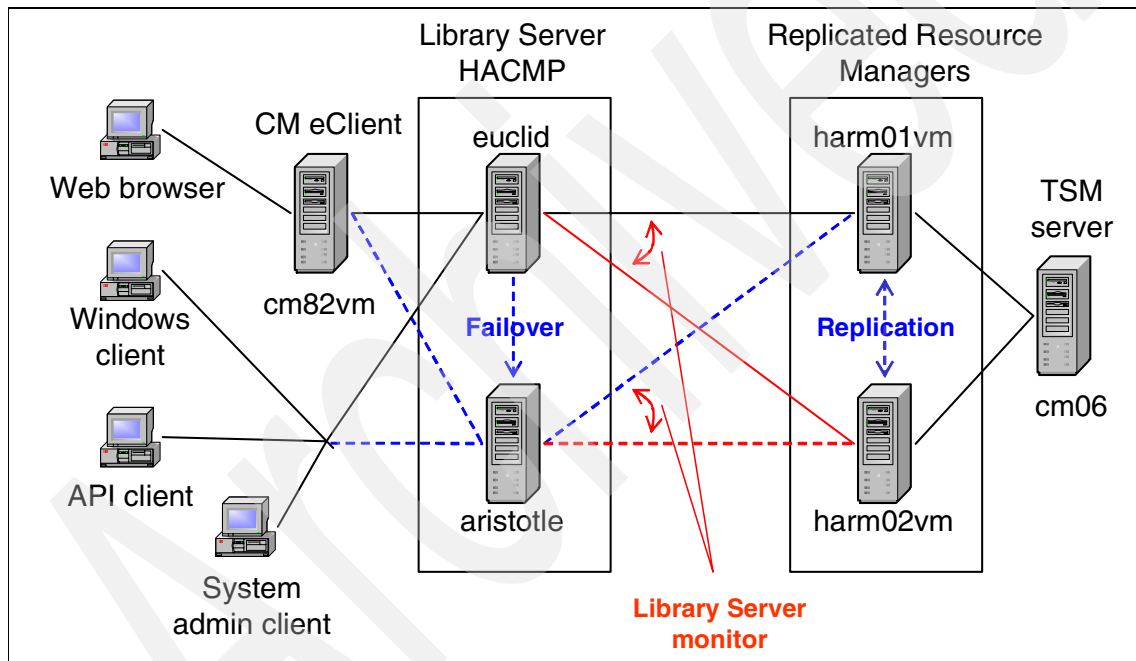


Figure 5-1 Our Content Manager high availability scenario

This scenario fits within the scope of 4.2.4, “High availability example 2: Clustering and replication” on page 110. Refer to that section if you want to learn more about how to compare this option with others available.

5.1.1 Out test case scenario

The scenario we built as a test case for this chapter consists of four main parts, of which the first two are configured to be highly available:

1. Library Server
2. Resource Managers
3. Tivoli Storage Manager server
4. Different components acting as clients to the Content Manager system:
 - Content Manager system administration interface
 - Content Manager Windows client
 - Content Manager eClient and Web browser
 - Custom Java API-based program

Library Server

For this component, we choose to use a platform clustering technique instead of DB2 replication, mainly for testing and documentation purposes. For the differences between these two options and the reasons that can lead to choose one or the other for a particular environment, refer to Chapter 4, “High availability strategies and options” on page 91.

The machines (euclid and aristotle) assigned for the Library Server installation are the two logical partitions in different IBM @server pSeries 690 systems. High availability is achieved using IBM High Availability Cluster Multi-Processing (HACMP) software. Figure 5-2 shows the physical configuration and cable setup.

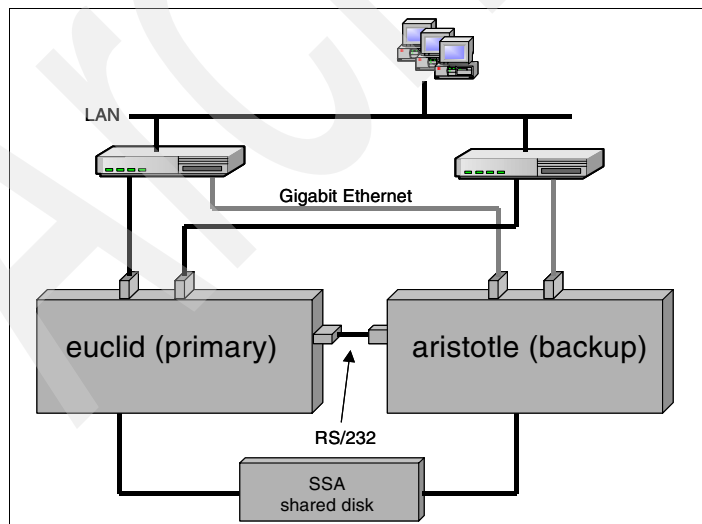


Figure 5-2 Library Server physical configuration and cable layout

Each partition has four processors and 16 GB of RAM and four 16 GB internal SCSI disks each. Also, the two partitions share four serial storage architecture (SSA) disks. Each machine also has one RS/232 and two Gigabit Ethernet adapters connected. This configuration meets all the requirements of an HACMP installation.

Using this base infrastructure, the Content Manager Library Server will be running on euclid, and HACMP will start it on aristotle in case of any failure. The main components that will be running in this cluster are the DB2 UDB server and the Content Manager Library Server monitor, responsible for the Resource Manager failover and fallback.

Resource Managers

For this component, we chose to use Content Manager replication over WebSphere cloning or hardware clustering. Again, the details about how to make a decision for a particular environment can be read in Chapter 4, “High availability strategies and options” on page 91.

Given that the nature of the high availability procedure we are using is platform independent, we chose to use Microsoft Windows as the platform for the Resource Managers to save installation and configuration time. We installed the primary Resource Manager in harm01vm and replicated every collection to a second Resource Manager installed in harm02vm. This can be seen in Figure 5-1 on page 122.

Both harm01vm and harm02vm are identical IBM @server xSeries® servers with 1 GB RAM, 1.2 GHz Intel® processor, and 8 GB of internal SCSI disk. Each Resource Manager is configured to connect to a Tivoli Storage Manager server installed on a separate box as a secondary storage medium.

The Library Server monitor mentioned in “Library Server” on page 123 is the component responsible to detect when the primary Resource Manager is down and to direct all further resource-related requests to the second Resource Manager. This component is also responsible to detect when the primary Resource Manager is up and resume normal operation for the entire system. This is further detailed in 5.3, “Resource Manager replication” on page 168.

Tivoli Storage Manager server

As shown in Figure 5-1 on page 122, cm06 has a Tivoli Storage Manager server Version 5.2 installed. It is configured to act as the Tivoli Storage Manager server for the entire installation and receives requests from both Resource Managers. This machine is an xSeries server with a 1.2 GHz Intel processor, 2 GB RAM, and 16 GB of storage space in one SCSI disk. The Tivoli Storage Manager storage configuration includes only disk pools.

In a real-world highly available scenario, this would not be the recommended topology because it introduces a single point of failure. In our test environment, the Tivoli Storage Manager component is simply assumed to be highly available by some means external to our environment and is present only to have a more complete setup from the Resource Manager component point of view.

Client applications

Refer back to Figure 5-1 on page 122; the last part of our environment consists of the different components that act as clients to the Content Manager system.

In one machine, we had the Content Manager System Administration Client, the Windows client, and a command-line Java application that uses the Content Manager APIs to load a new image to a particular item type every three seconds.

We also have the Content Manager eClient mid-tier server running on a separate box called cm82vm. This client is accessed through a Web browser in a different machine.

The software products are at the same following version and fix pack level in each component of the installation:

- ▶ DB2 UDB Version 8.1 Fix Pack 3
- ▶ WebSphere Application Server Version 5.0 Fix Pack 2
- ▶ Content Manager Version 8.2 Fix Pack 2

5.2 Library Server HACMP

High availability for the Library Server component is achieved by using IBM High Availability Cluster Multi-Processing (HACMP). HACMP for AIX provides a highly available computing environment. This facilitates the automatic switching of users, applications, and data from one system in the cluster to another after a hardware or software failure. A complete high availability (HA) setup includes many parts, one of which is the HACMP software. Other parts of an HA solution come from AIX and the logical volume manager (LVM).

In HACMP, the components that can be automatically switched between different nodes in the cluster can include hardware and software, such as:

- ▶ Disks
- ▶ Volume groups
- ▶ File systems
- ▶ Network addresses

- ▶ Application servers
- ▶ SCSI or Fibre Channel tape drives

In order to make HACMP take control of these resources, they must be grouped into a “resource group.” With this, you can combine the different parts that you want to protect and define the relationship between the different components of an HA installation, such as the member nodes or takeover relationships. Takeover relationships for resource groups can be defined in many ways (cascading, rotating, or concurrent), but this is not within the scope of this redbook. To learn more about resource groups and takeover relationships, refer to Chapter 1, “Cluster Resources and Resource Groups,” in *Concepts and Facility Guide for HACMP Version 4.5*, SC23-4276.

In our test environment, we define a hot standby with a cascading without fallback resource group for the Library Server component. This means that if the owner node of this resource group (primary node, euclid) experiences a failure, the other node (backup node, aristotle) will take care of the IP address, file systems, and Library Server application. When the other node rejoins the cluster (start HACMP daemons), the Library Server will remain on the backup and will not automatically fall back to the other node. Refer to *Concepts and Facility Guide for HACMP Version 4.5*, SC23-4276, for a detailed explanation.

Following are some of the terms that are used in this chapter:

PV	Physical volume
VG	Volume group
LV	Logical volume
JFSLog	Journalized file system log
JFS	Journalized file system
hdisk	Logical name of a disk device
LS	Library Server
RM	Resource Manager
euclid and aristotle	The name of the nodes on the cluster
Primary node	The one that will run the LS (euclid)
Backup/standby node	The one that will be the target for the LS failover (aristotle)

5.2.1 Software installation

Before doing any configuration, all the required software must be installed in both nodes of the cluster. For a list of required software, refer to Chapter 17, “Installing

and updating prerequisite programs for AIX”, in *Planning and Installing Your Content Management System*, GC27-1332. Following is a list of the software, which is installed in both euclid and aristotle for our test environment:

1. AIX 5L Version 5.2 with Maintenance Level 2
2. HACMP Classic 4.5
3. VisualAge® C++ for AIX Version 5.0.2
4. DB2 UDB Version 8.1 with Fix Pack 3
5. Content Manager server Version 8.2 with Fix Pack 2

These products should be installed and their fix packs applied, but no further configuration should be done. Do not create a DB2 instance and do not create any Content Manager Library Server or Resource Manager databases. These configuration steps are explained in the following sections.

5.2.2 Library Server parameter values

Before performing any configuration step, we must decide the values of different parameters needed for the installation of the Library Server. Most of these steps are explained in detail in *Planning and Installing Your Content Management System*, GC27-1332. We highly recommend that the configuration steps specified in that publication are thoroughly known by you before attempting to follow any of the configuration steps in this redbook.

For our setup, we decided to use ICMHADB as the name of the Library Server database. We decided not to use Content Manager text-search, any kind of LDAP integration, or single sign-on.

Other parameters that have to be defined for this environment are the AIX user IDs, groups, and file systems that have to be created. These must be precisely identified and documented for you to be able to perform an appropriate HACMP definition. Otherwise, it is very probable that problems will be found during testing or HACMP failover. Table 5-1 and Table 5-2 on page 128 list the groups and user IDs that have to be defined.

Table 5-1 Library Server AIX groups

Description	Group name
DB2 instance owners group	db2iadm1
DB2 fenced ID primary group	db2fgrp

Table 5-2 Library Server AIX user IDs

Description	User name
DB2 instance owner	db2inst1
DB2 fenced ID	db2fenc1
CM Library Server administrator	icmadmin
CM database connection ID	icmconct

Resource Manager user IDs are not necessary given that we will only configure the Library Server component at this time.

Next, the file system layout has to be defined. It is important to keep in mind at the same time the considerations regarding file system separation for Content Manager performance purposes and the file system sharing layout for HACMP failover.

For information about performance considerations, we recommend the IBM Redbook *Performance Tuning for Content Manager*, SG24-6949 and the white paper *IBM Content Manager v8.2 Performance Tuning Guide*. To review the white paper, go to the following URL, select **Learn**, and enter the search string performance tuning guide:

<http://www.ibm.com/software/data/cm/cmgr/mp/support.html>

In our test scenario, we decided to use:

- ▶ A single separate file system for the Library Server database tablespace containers
- ▶ A single separate file system for the Library Server database active logs

We identified the file system hierarchy locations where there is data that is important to the Content Manager system and rate it according to the frequency with which this information changes. This is summarized in Table 5-3.

Table 5-3 File system locations with Library Server data

Default location	Description	Update frequency
/home/db2inst1/	DB2 instance owner home	Low
/home/db2inst1/sqlib	DB2 instance home	Medium
/home/db2inst1/db2inst1	DB2 default database directory	High

Default location	Description	Update frequency
/home/db2inst1/db2inst1/ NODE0000/SQL00001/SQLOGDIR	Library Server database log directory	High
/home/db2fenc1	DB2 fenced ID home	Low
/home/db2fenc1/ICMLNSDB/DLL	CM generated access modules	Medium
/home/icmadmin	CM administrator home	Low
/home/icmconct	CM DB connect ID home	Low

In order for the backup node to be able to pick up the Library Server work in case of a failure of the primary node, it has to be able access all the data that the primary node was using at the time of the failure. This can be done by either having an identical file system on both nodes or by using a storage device that can be accessed by both nodes and switching it from one node to the other when the failure occurs. The last option is called *shared file systems*.

Using the information in Table 5-3 on page 128 and the mentioned considerations, we define the file systems for our Library Server installation in Table 5-4.

Table 5-4 Library Server shared file systems

Description	File system location
DB2 instance directory and owner's home	/home/dbinst1
Library Server tablespace containers	/udbdata/icmhadb
Library Server database logs	/udblogs/icmhadb
DB2 fenced ID's home and Library Server generated access modules	/home/db2fenc1

Both the Library Server administrator's home and the database connection ID's home will be created on the /home file system and will not be shared between nodes. Instead, the configuration files that reside on them will be kept the same by manually copying them each time they change.

5.2.3 File system setup and user IDs creation

Using the information gathered in the previous section, we need to define all the required groups, user IDs, and file systems. All these definitions and steps must

be run on the primary node, because we need to set up this first in order to continue with the configuration of the Content Manager components.

Before starting with the definition of the users, we must create the file systems involved on the installation and mount them. For our example, we use the file system layout as specified in Table 5-5.

Table 5-5 File systems and LVM layout for the Library Server

File system name	Volume group name	Logical volume name	Physical volume name
/home/db2inst1	vgdb2bin	lvdb2inst1	hdisk4
/home/db2fenc1	vgdb2data	lvdb2fenc1	hdisk5 hdisk6
/udbdata/icmhadb	vgdb2data	lvdb2data	hdisk5 hdisk6
/udblogs/icmhadb	vgdb2logs	lvdb2logs	hdisk7

We define the volume groups, logical volumes, and file systems first on euclid and then import them into aristotle.

To create the volume groups, contact the system administrator to know which disks are available to use. We use the following procedure:

1. Create a volume group.

You can accomplish this with smit:

- a. Enter:

```
# smit mkvg
```

- b. Enter the appropriate values, as shown in Figure 5-3 on page 131.

Add a Volume Group

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

	[Entry Fields]	
VOLUME GROUP name	[vgdb2bin]	
Physical partition SIZE in megabytes	32	+
* PHYSICAL VOLUME names	[hdisk4]	+
Force the creation of a volume group?	no	+
Activate volume group AUTOMATICALLY at system restart?	no	+
Volume Group MAJOR NUMBER	[]	+#
Create VG Concurrent Capable?	no	+
Create a big VG format Volume Group?	no	+
LTG Size in kbytes	128	+

Figure 5-3 Add a volume group smit panel

Alternatively, you can use the following command line:

```
# mkvg -y'vgdb2bin' -s'32' '-n' hdisk4
```

For the other volume groups, do the following:

```
# mkvg -y'vgdb2data' -s'32' '-n' hdisk5 hdisk6
```

```
# mkvg -y'vgdb2logs' -s'32' '-n' hdisk7
```

Note: Make sure you select *not to* activate the volume group at system restart flag (flag -n); HACMP will take care of the activation.

2. Create the JFS log and logical volumes (LVs).

When you create a JFS with smit, AIX creates the JFS log automatically with a defined standard name such as loglv00 and loglv01. This is same for the logical volumes but with names such as lv00 and lv01. To avoid having the same name for LVs or JFSLogs, it is a good practice to create the logical volumes first, so you can choose the name for them.

To create the JFSLogs (you need at least one per volume group and per JFS) and the logical volumes that will hold the data, use any name that is not already used on the nodes that are involved. Remember that the file systems

will be available on all the nodes, and you cannot have duplicated names for logical volumes in the same machine.

a. Create the logical volumes.

You can accomplish this with smit:

i. Enter:

```
# smit mklv
```

ii. Type the appropriate volume group name and other values, as shown in Figure 5-4.

Add a Logical Volume

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]	[Entry Fields]	
Logical volume NAME	[loglvdb2bin]	
* VOLUME GROUP name	vgdb2bin	
* Number of LOGICAL PARTITIONS	[1]	#
PHYSICAL VOLUME names	[]	+
Logical volume TYPE	[jfslog]	+
POSITION on physical volume	outer_middle	+
RANGE of physical volumes	minimum	+
MAXIMUM NUMBER of PHYSICAL VOLUMES	[]	#
to use for allocation		
Number of COPIES of each logical partition	1	+
Mirror Write Consistency?	active	+
Allocate each logical partition copy on a SEPARATE physical volume?	yes	+
RELOCATE the logical volume during reorganization?	yes	+
Logical volume LABEL	[]	
MAXIMUM NUMBER of LOGICAL PARTITIONS	[512]	#
Enable BAD BLOCK relocation?	yes	+
SCHEDULING POLICY for reading/writing logical partition copies	parallel	+
Enable WRITE VERIFY?	no	+
File containing ALLOCATION MAP	[]	
Stripe Size?	[Not Striped]	+
Serialize IO?	no	+

Figure 5-4 Creation of a JFSLog logical volume

Alternatively, you can use the following command line:

```
# mklv -y'loglvdb2bin' -t'jfslog' vgdb2bin 1
```

b. Create the JFS logical volumes.

You can accomplish this with smit:

i. Enter:

```
# smit mklv
```

ii. Type the appropriate volume group name and enter other values, as shown in Figure 5-5.

Add a Logical Volume

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]	[Entry Fields]	
Logical volume NAME	[lvdb2inst1]	
* VOLUME GROUP name	vgdb2bin	
* Number of LOGICAL PARTITIONS	[64]	#
PHYSICAL VOLUME names	[] +	
Logical volume TYPE	[jfs]	+
POSITION on physical volume	outer_middle	+
RANGE of physical volumes	minimum	+
MAXIMUM NUMBER of PHYSICAL VOLUMES to use for allocation	[]	#
Number of COPIES of each logical partition	1	+
Mirror Write Consistency?	active	+
Allocate each logical partition copy on a SEPARATE physical volume?	yes	+
RELOCATE the logical volume during reorganization?	yes	+
Logical volume LABEL	[]	
MAXIMUM NUMBER of LOGICAL PARTITIONS	[512]	#
Enable BAD BLOCK relocation?	yes	+
SCHEDULING POLICY for reading/writing logical partition copies	parallel	+
Enable WRITE VERIFY?	no	+
File containing ALLOCATION MAP	[]	
Stripe Size?	[Not Striped]	+
Serialize IO?	no	+

Figure 5-5 Creation of a JFS logical volume

Alternatively, you can use the following command line:

```
# mklv -y'lvdb2inst1' -t'jfs' vgdb2bin 64
```

c. Create the remaining logs and logical volumes using the following:

```
# mklv -y'loglvdb2data' -t'jfslog' vgdb2data 1
```

```
# mklv -y'loglvdb2logs' -t'jfslog' vgdb2logs 1
# mklv -y'lvdb2data' -t'jfs' vgdb2data 64
# mklv -y'lvdb2fenc1' -t'jfs' vgdb2data 32
# mklv -y'lvdb2logs' -t'jfs' vgdb2logs 64
```

- d. Format the JFSLogs by issuing the **logform** command with each JFSLog created and answer “y” when asked to destroy the LV. See Example 5-1 on page 122.

Example 5-1 Format of a JFSLog volume

```
# logform /dev/loglvdb2bin
logform: destroy /dev/loglvdb2bin (y)? y
```

- e. Verify what you have done by issuing the following command:

```
lsvg -l <vg name>
```

3. Create the file systems.

After the LVs have been created, we are ready to create and add a file system associated with those LVs. We create a “Large File Enabled Journaled File System” to accommodate the potential future database file growth that might exceed 2 GB, the limit of a standard file system.

You can accomplish this with smit:

- a. Enter:
- ```
smit jfs
```
- b. Select **Add a Journaled File System on a Previously Defined Logical Volume → Add a Large File Enabled Journaled File System**.
- c. Enter the values as shown in Figure 5-6 or as appropriate for your system.

Add a Large File Enabled Journaled File System

Type or select values in entry fields.  
Press Enter AFTER making all desired changes.

|                                        | [Entry Fields]   |   |
|----------------------------------------|------------------|---|
| * LOGICAL VOLUME name                  | lvdb2inst1 +     |   |
| * MOUNT POINT                          | [/home/db2inst1] |   |
| Mount AUTOMATICALLY at system restart? | no               | + |
| PERMISSIONS                            | read/write       | + |
| Mount OPTIONS                          | []               | + |
| Start Disk Accounting?                 | no               | + |
| Number of bytes per inode              | 4096             | + |
| Allocation Group Size (MBytes)         | 64               | + |

Figure 5-6 Add a Large File Enabled Journal File System smit panel

Alternatively, you can use the following command:

```
/usr/sbin/crfs -v jfs -a bf=true -d'lvdb2inst1' -m'/home/db2inst1'\
-A''locale nostr | awk -F: '{print $1}' -p'rw' -t''locale nostr | awk
-F:\ '{print $1}' -a n bpi='4096' -a ag='64'
```

For the other file systems, use the following commands:

```
/usr/sbin/crfs -v jfs -a bf=true -d'lvdb2fenc1' -m'/home/db2fenc1'\
-A''locale nostr | awk -F: '{print $1}' -p'rw' -t''locale nostr | awk
-F:\ '{print $1}' -a n bpi='4096' -a ag='64'
/usr/sbin/crfs -v jfs -a bf=true -d'lvdb2data' -m'/udbdata/icmhadb'\
-A''locale nostr | awk -F: '{print $1}' -p'rw' -t''locale nostr | awk
-F:\ '{print $1}' -a n bpi='4096' -a ag='64'
/usr/sbin/crfs -v jfs -a bf=true -d'lvdb2logs' -m'/udblogs/icmhadb'\
-A''locale nostr | awk -F: '{print $1}' -p'rw' -t''locale nostr | awk
-F:\ '{print $1}' -a n bpi='4096' -a ag='64'
```

The only thing that you need to pay attention to here is to be sure that the LV that you select matches the mount point that you choose for it. In our configuration, the logical volume to mount point relationship is specified in Table 5-6.

Table 5-6 Logical volume to mount point relationship

| Logical volume | Mount point/File system |
|----------------|-------------------------|
| lvdb2inst1     | /home/db2inst1          |
| lvdb2fenc1     | /home/db2fenc1          |
| lvdb2data      | /udbdata/icmhadb        |
| lvdb2logs      | /udblogs/icmhadb        |

Follow the same step for all other file systems (in our scenario, **/home/db2fenc1**, **/udbdata/icmhadb** and **/udblogs/icmhadb**).

Mount them by issuing:

```
mount /home/db2inst1
mount /home/db2fenc1
mount /udbdata/icmhadb
mount /udblogs/icmhadb
```

#### 4. Create the group and user IDs.

Now that the file systems are created and mounted, we can continue with the creation of the users and groups.

**Note:** Before starting with the group and user creation, make sure that the volume groups are activated and the file systems involved are mounted.

**Important:** Keep in mind that the IDs for the users involved with DB2 and Content Manager must be *exactly the same* between the nodes that compose the cluster, because if a failover occurs, we need to have the same IDs on the target node in order to have the Library Server work properly. See Chapter 1, “Defining the Takeover Relationships Among Cluster Nodes,” in *Concepts and Facilities Guide for HACMP 4.5*, SC23-4276.

On *both* nodes, issue the following commands. Note that the ID values that we choose are the same. The users that we define are the ones specified in 5.2.2, “Library Server parameter values” on page 127.

a. Create the groups with these commands:

```
mkgroup id=1000 db2iadm1
mkgroup id=1001 db2fgrp
```

b. Create the users and set the password with these commands:

```
mkuser id='2000' pgrp='db2iadm1' groups='db2iadm1'
home='/home/db2inst1'\ db2inst1
mkuser id='2001' pgrp='db2fgrp' groups='db2fgrp'
home='/home/db2fenc1'\ db2fenc1
mkuser id='2002' pgrp='db2iadm1' groups='db2iadm1'
home='/home/icmadmin'\ icmadmin
mkuser id='2003' icmcont
passwd db2inst1
passwd db2fenc1
passwd icmadmin
passwd icmcont
```

For more information about the **mkgroup**, **passwd**, or **mkuser** commands, refer to the AIX documentation or man pages.

5. Change ownership and permission.

For the DB2 and Content Manager components to be able to write to the file systems that we defined, we have to change the ownership. These commands must be run on the node on which the file systems are already defined and mounted. Use the following commands:

```
chown db2inst1:db2iadm1 /udbdata/icmhadb
chmod 755 /udbdata/icmhadb
```

Also, the permissions for all the directories in the directory path up to the mount point for these file systems should be as the following example:

```
drwxr-xr-x 3 db2inst1 db2iadm1 256 Nov 04 14:47 /udbdata
```

Issue the **chown** and **chmod** commands for every directory in each of the nodes where these file systems will be mounted. Any error on this permissions setup can lead to unpredictable results in the case of a Library Server failover.

**Important:** Any change to user or group IDs, passwords, or limits on the users that are involved in a HACMP configuration must be reflected on *all* the nodes that are on the cluster.

## 5.2.4 DB2 and Content Manager instance creation on primary node

**Note:** Do not put any DB2 or Library Server start or stop script on `/etc/inittab`, because when the server boots, the file systems will not be mounted and some services are not started. After the HACMP daemons start, these resources will be available and ready to use. The start and stop scripts will be managed by an *application server*. It is the cluster resource that manages the applications.

After all the file systems and users are created, it is time to perform all the required configuration steps to have a ready to use Library Server in this node. Before continuing with this part of the redbook, we highly recommend that you read Chapter 18, “Performing pre-installation steps on AIX,” and Chapter 40, “Using Content Manager after-install programs and procedures,” in *Planning and Installing Your Content Management System*, GC27-1332.

In our environment, user IDs and file systems are already created, but there is still no DB2 instance. So, the work to be done consists of the creation and customization of a DB2 instance, the configuration of environment settings for AIX users, and the creation of a new Library Server database as follows:

1. Create DB2 instance.

Having the file systems mounted on the primary node, log in as root and create a DB2 instance using the following command:

```
/usr/opt/db2_08_01/instance/db2icrt -u db2fenc1 db2inst1
```

This creates the DB2 instance on the `/home/db2inst1` directory with default settings. `db2inst1` will be the owner of the instance and `db2fenc1` the fenced ID. In many cases, you will probably want to create a DB2 administrative instance as well. As this was not the case in our sample scenario, this is not covered in this book.

Verify that the TCP/IP port information has been correctly stored in the `/etc/services` file. Example 5-2 on page 138 is a fragment of the services file stored in `euclid` after the instance creation.

### Example 5-2 Services file in euclid

---

```
DB2_db2inst1 60000/tcp
DB2_db2inst1_1 60001/tcp
DB2_db2inst1_2 60002/tcp
DB2_db2inst1_END 60003/tcp
```

---

Take note of the port number used by DB2 for TCP/IP communications.

#### 2. Configure the DB2 default path for the database location.

In 5.2.2, “Library Server parameter values” on page 127, we define that the database data files will be located in the /udbdata/icmhadb directory. This can be done by specifying an alternate location for the database during database creation or by altering the default database path for the DB2 instance. Given that the database creation is automatically done by Content Manager scripts, we recommend using the second approach.

To do this, simply **su** to the instance owner user ID and change the database manager variable DFTDBPATH to the desired database data file directory:

```
su - db2inst1
$ db2 update dbm cfg using DFTDBPATH /udbdata/icmhadb
```

You can verify that the configuration took place by issuing:

```
$ db2 get dbm cfg
```

Looking for the value of the DB2DFTPATH variable. Example 5-3 is the output in our system. The DB2DFTPATH information is in bold text.

### Example 5-3 Database manager configuration for euclid

---

#### Database Manager Configuration

Node type = Enterprise Server Edition with local and remote clients

Database manager configuration release level = 0x0a00

CPU speed (millisec/instruction) (CPUSPEED) = 4.684080e-07

Communications bandwidth (MB/sec) (COMM\_BANDWIDTH) = 1.000000e+02

Max number of concurrently active databases (NUMDB) = 8

Data Links support (DATA LINKS) = NO

Federated Database System Support (FEDERATED) = NO

Transaction processor monitor name (TP\_MON\_NAME) =

Default charge-back account (DFT\_ACCOUNT\_STR) =

Java Development Kit installation path (JDK\_PATH) = /usr/java131



|                                |                                                |
|--------------------------------|------------------------------------------------|
| Diagnostic error capture level | (DIAGLEVEL) = 3                                |
| Notify Level                   | (NOTIFYLEVEL) = 3                              |
| Diagnostic data directory path | (DIAGPATH) = /home/db2inst1/<br>sqllib/db2dump |

#### Default database monitor switches

|                                          |                           |
|------------------------------------------|---------------------------|
| Buffer pool                              | (DFT_MON_BUFPOOL) = OFF   |
| Lock                                     | (DFT_MON_LOCK) = OFF      |
| Sort                                     | (DFT_MON_SORT) = OFF      |
| Statement                                | (DFT_MON_STMT) = OFF      |
| Table                                    | (DFT_MON_TABLE) = OFF     |
| Timestamp                                | (DFT_MON_TIMESTAMP) = OFF |
| Unit of work                             | (DFT_MON_UOW) = OFF       |
| Monitor health of instance and databases | (HEALTH_MON) = OFF        |

|                     |                           |
|---------------------|---------------------------|
| SYSADM group name   | (SYSADM_GROUP) = DB2IADM1 |
| SYSCTRL group name  | (SYSCTRL_GROUP) =         |
| SYSMAINT group name | (SYSMAINT_GROUP) =        |

|                                      |                           |
|--------------------------------------|---------------------------|
| Database manager authentication      | (AUTHENTICATION) = SERVER |
| Cataloging allowed without authority | (CATALOG_NOAUTH) = NO     |
| Trust all clients                    | (TRUST_ALLCLNTS) = YES    |
| Trusted client authentication        | (TRUST_CLNTAUTH) = CLIENT |
| Bypass federated authentication      | (FED_NOAUTH) = NO         |

|                              |                                       |
|------------------------------|---------------------------------------|
| <b>Default database path</b> | <b>(DFTDBPATH) = /udbdata/icmhadb</b> |
|------------------------------|---------------------------------------|

|                                      |                               |
|--------------------------------------|-------------------------------|
| Database monitor heap size (4KB)     | (MON_HEAP_SZ) = 90            |
| Java Virtual Machine heap size (4KB) | (JAVA_HEAP_SZ) = 2048         |
| Audit buffer size (4KB)              | (AUDIT_BUF_SZ) = 0            |
| Size of instance shared memory (4KB) | (INSTANCE_MEMORY) = AUTOMATIC |
| Backup buffer default size (4KB)     | (BACKBUFSZ) = 1024            |
| Restore buffer default size (4KB)    | (RESTBUFSZ) = 1024            |

|                           |                      |
|---------------------------|----------------------|
| Sort heap threshold (4KB) | (SHEAPTHRES) = 10000 |
|---------------------------|----------------------|

|                         |                   |
|-------------------------|-------------------|
| Directory cache support | (DIR_CACHE) = YES |
|-------------------------|-------------------|

|                                           |                         |
|-------------------------------------------|-------------------------|
| Application support layer heap size (4KB) | (ASLHEAPSZ) = 15        |
| Max requester I/O block size (bytes)      | (RQRIOBLK) = 32767      |
| Query heap size (4KB)                     | (QUERY_HEAP_SZ) = 32768 |
| DRDA services heap size (4KB)             | (DRDA_HEAP_SZ) = 128    |

|                                        |                         |
|----------------------------------------|-------------------------|
| Workload impact by throttled utilities | (UTIL_IMPACT_LIM) = 100 |
|----------------------------------------|-------------------------|

|                                  |                                    |
|----------------------------------|------------------------------------|
| Priority of agents               | (AGENTPRI) = SYSTEM                |
| Max number of existing agents    | (MAXAGENTS) = 500                  |
| Agent pool size                  | (NUM_POOLAGENTS) = 250(calculated) |
| Initial number of agents in pool | (NUM_INITAGENTS) = 0               |

|                                           |                                                  |
|-------------------------------------------|--------------------------------------------------|
| Max number of coordinating agents         | (MAX_COORDAGENTS) = (MAXAGENTS - NUM_INITAGENTS) |
| Max no. of concurrent coordinating agents | (MAXCAGENTS) = MAX_COORDAGENTS                   |
| Max number of client connections          | (MAX_CONNECTIONS) = MAX_COORDAGENTS              |
| Keep fenced process                       | (KEEPFENCED) = YES                               |
| Number of pooled fenced processes         | (FENCED_POOL) = MAX_COORDAGENTS                  |
| Initialize fenced process with JVM        | (INITFENCED_JVM) = NO                            |
| Initial number of fenced processes        | (NUM_INITFENCED) = 0                             |
| Index re-creation time                    | (INDEXREC) = RESTART                             |
| Transaction manager database name         | (TM_DATABASE) = 1ST_CONN                         |
| Transaction resync interval (sec)         | (RESYNC_INTERVAL) = 180                          |
| SPM name                                  | (SPM_NAME) = euclid                              |
| SPM log size                              | (SPM_LOG_FILE_SZ) = 256                          |
| SPM resync agent limit                    | (SPM_MAX_RESYNC) = 20                            |
| SPM log path                              | (SPM_LOG_PATH) =                                 |
| TCP/IP Service name                       | (SVCENAME) = DB2_db2inst1                        |
| Discovery mode                            | (DISCOVER) = SEARCH                              |
| Discover server instance                  | (DISCOVER_INST) = ENABLE                         |
| Maximum query degree of parallelism       | (MAX_QUERYDEGREE) = ANY                          |
| Enable intra-partition parallelism        | (INTRA_PARALLEL) = NO                            |
| No. of int. communication buffers(4KB)    | (FCM_NUM_BUFFERS) = 4096                         |
| Node connection elapse time (sec)         | (CONN_ELAPSE) = 10                               |
| Max number of node connection retries     | (MAX_CONNRETRIES) = 5                            |
| Max time difference between nodes (min)   | (MAX_TIME_DIFF) = 60                             |
| db2start/db2stop timeout (min)            | (START_STOP_TIME) = 10                           |

---

### 3. Update DB2 and user profiles.

- a. In the .profile file of the icmadmin home directory, add the following lines, which can be copied from the same file in the home directory of the DB2 instance owner:

```
if [-f /home/db2inst1/sqllib/db2profile]; then
 . /home/db2inst1/sqllib/db2profile
fi
```

Replace /home/db2inst1 with your instance owner's home directory.

- b. Edit the DB2 instance profile.env file, which should be on the sqllib directory, under the DB2 instance owner's home directory, by adding the following two lines at the beginning:

```
DB2LIBPATH=/usr/lib:/usr/lpp/icm/lib
DB2ENVLIST='LIBPATH ICMROOT ICMDLL ICMCOMP EXTSHM CMCOMMON'
```

- c. Edit the user profile, which should be in the same directory, by adding the following information:

```
export ICMCOMP=/usr/vacpp/bin
export ICMROOT=/usr/lpp/icm
export EXTSHM=ON
export ICMDLL=/home/db2fenc1
export INCLUDE=/home/db2inst1/sqllib/include:$INCLUDE
export LIBPATH=$ICMROOT/lib:$LIBPATH
export CMCOMMON=/usr/lpp/cmb/cmgt
export PATH=$PATH:$ICMROOT/bin/DB2
```

If this file does not exist, create a new one. After this step is completed, all the environment should be ready for the Library Server database to be created and to be functional.

4. Create the Content Manager Library Server database.

- a. Login as root and establish the DB2 environment with the following command:

```
./home/db2inst1/sqllib/db2profile
```

Pay attention to the space between the . sign and the / sign.

- b. Change to the Content Manager configuration directory:

```
cd /usr/lpp/icm/config
```

- c. Run the Content Manager Library Server creation utility:

```
./icmcreatesdb.sh
```

Reply to every prompt using the information you gathered in 5.2.2, “Library Server parameter values” on page 127. In our case, Example 5-4 shows what happened when we ran the utility.

*Example 5-4 Library Server parameter summary*

---

```
Database name: ICMHADB
Replace existing database: Yes
Database connection ID: ICMCONCT
LS database administrator ID: ICMADMIN
Schema name: ICMADMIN
Library Server ID: 1
Path into which the Library Server was installed: /usr/lpp/icm
Path to be used for system generated DLLs: /home/db2fenc1
Enable Unicode support: No
```

Enable net search support: No  
Token duration time in hours: 48  
Resource Manager host name:: harm01vm.svl.ibm.com  
Resource Manager http port number: 80  
Resource Manager https port number: 443  
Resource Manager database name: HARMDB01  
Resource Manager access ID: RMADMIN  
Resource Manager Web application path: /icrmr  
Resource Manager platform: Windows  
Host name: euclid.svl.ibm.com  
Port number: 60000  
Node number:  
Enable SSO support: No  
Server authentication: No

---

Notice that the database name, host name, and port numbers of the first Resource Manager should be used. In our scenario, HARMDB01 is the database name of the first Resource Manager. Refer to 5.3, “Resource Manager replication” on page 168 for further details about the Resource Manager setup. Also, use the port number that was present in the /etc/services file in the Port Number entry.

After some processing time, you should see messages confirming the successful creation of the Library Server database, including the access module generation for standard Content Manager item types. If you run into any problems, follow the normal problem solving process as you would follow in any standard Library Server installation.

## 5.2.5 Setting up shared disks and LVM

In this section, we go through the steps to configure the volume groups and file systems that were defined on *euclid* so they will be available on *aristotle* if a take over happens or if you decide to move the resource group to the other node manually.

First of all, we must clarify that this redbook is not intended to cover the requirements, configuration steps, or best practices to set up shared disks. We also do not cover any performance-related issues or recommendations. This is because the need could be very different in each particular installation. We only follow the steps that we need to accomplish the tasks for the high availability configuration in our sample environment.

For our scenario, assuming that the same disks can be accessed through the nodes participating on the cluster, we do the steps to configure and set up the disks, volume groups, and file systems to continue with the HACMP configuration.

**Note:** You might need to ask your system or storage administrator for this, because this could involve several devices or configuration steps.

To continue with the configuration, we need to import on the secondary node the volume groups that were defined on the primary node.

To make the data available on node *aristotle*, complete the following steps:

1. Unmount the file systems and deactivate the volume groups on primary node.

The first thing that we need to do is stop all the applications, unmount the file systems, and deactivate the volume groups. In order to do this, the Library Server must be down, including the Library Server monitor and the DB2 database manager. If you have your Library Server running, use the following two procedures to stop it.

- To stop the Library Server monitor, log in as root and run:

```
/etc/rc.cmlsproc -shutdown
```

- To stop the DB2 database manager, log in as **db2inst1** and run:

```
db2stop force
```

This way, we already have defined the file systems, created users, and configured the Library Server in *euclid*, so we need to continue on this node.

- a. Unmount the file systems:

```
unmount /udblogs/icmhadb
unmount /udbdata/icmhadb
unmount /home/db2fenc1
unmount /home/db2inst1
```

**Tip:** If the following message appears:

```
root@euclid # unmount /home/db2inst1
umount: 0506-349 Cannot unmount /dev/lvdb2inst1: The requested
resource is busy.
```

This means that at least one process is using a file or a file structure. Make sure that the Library Server has been shut down and try again. You could run the **fuser -u /dev/<lv name>** command to see which process is still using the file system.

- b. Deactivate all the file systems on the volume groups:

```
varyoffvg vgdb2data
varyoffvg vgdb2logs
varyoffvg vgdb2bin
```

Remember to keep a terminal session open to this node.

2. Import the VG on the backup node.

a. Log in to the backup node.

The first thing to do is to import the volume groups from the other node to get the data available in case a takeover occurs. To accomplish that, we need to know at least one disk member of the volume group that we are trying to define.

An important thing to note is that the physical volume name could be different between the servers. One way to identify the physical volume is to match the physical volume ID (*PVID*) of the disk. If you run the command **lspv** on the node on which you already have the volume groups defined (primary node), you will notice that the second column contains a 16 character mix of numbers and letters, that's the PVID. See Example 5-5.

*Example 5-5 Output of lspv command on the primary node*

---

|               |                  |           |        |
|---------------|------------------|-----------|--------|
| root@euclid # | lspv             |           |        |
| hdisk0        | 0020390ab93b636f | rootvg    | active |
| hdisk1        | 0020390a7f2f4639 | rootvg    | active |
| hdisk2        | 0020390a6f91263e | pagevg    | active |
| hdisk3        | 0020390a7f2f2533 | pagevg    | active |
| hdisk4        | 0020390aa801dfef | vgdb2logs | active |
| hdisk5        | 0020390aa802b6ae | vgdb2data | active |
| hdisk6        | 0020390aa802bcda | vgdb2data | active |
| hdisk7        | 0020390aa8097138 | vgdb2bin  | active |

---

It is possible that on the other node the PVIDs are shown as None, because perhaps the disks have never been accessed or the PVID was removed. To read the PVID from the disk and store the information on the ODM, run this command:

```
chdev -l hdiskXXX -a pv='yes'
```

Where **hdiskXXX** is a valid physical volume name. See Example 5-6.

*Example 5-6 Obtaining the PVID from a physical disk*

---

|                  |                              |      |  |
|------------------|------------------------------|------|--|
| root@aristotle # | lspv   grep hdisk28          |      |  |
| hdisk28          | none                         | None |  |
| root@aristotle # | chdev -l hdisk28 -a pv='yes' |      |  |
| hdisk28          | changed                      |      |  |
| root@aristotle # | lspv   grep hdisk28          |      |  |
| hdisk28          | 0020390ac7d3a652             | None |  |

---

**Note:** Try to run this command on the disks that have the None value in the second column. If you are not very sure about this, ask your system administrator about how to map the disk of the first node to the second one.

Now, with the PVID set in the secondary node, we need to uniquely identify each disk. To do that, run the **lspv** command on the first node and write down or record the PVID number of one member of the volume group that you need to import. Example 5-7 shows how you can do that.

*Example 5-7 How to get one PVID for a volume group*

```
root@euclid # lspv | grep vgdb2data | head -n1
hdisk5 0020390aa802b6ae vgdb2data active
root@euclid #
```

On the secondary node, run the **lspv** command and identify the physical volume name that matches the PVID from the first node. Example 5-8 shows how you can do this.

*Example 5-8 Finding the PVID*

```
root@aristotle # lspv | grep "0020390aa802b6ae"
hdisk41 0020390aa802b6ae None
root@aristotle #
```

The third column from the output represents the volume group to which the disk is assigned. The disks that we are going to work will be have the None value because they are not assigned yet. If this column shows another value, consult your system administrator.

b. Import the VG.

With this above information, we can run the **importvg** command on the other node using this syntax:

```
importvg -y <VG name> <disk member>
```

Example 5-9 shows how you can do this.

*Example 5-9 Importing a volume group*

```
root@aristotle # importvg -y vgdb2data hdisk41
vgdb2data
root@aristotle #
```

c. You need to do this with *all* the volume groups involved (in our case, vgdb2data, vgdb2logs, and vgdb2bin).

**Note:** In our scenario, we did not set up the Major Number because we are not using NFS with HACMP. If you are using NFS within the scope of HACMP, you need to import the volume group matching this number with the one on the other node. For a further explanation of this and reference information, see Chapter 5, “Additional AIX Administrative Tasks,” in *Installation Guide for HACMP V4.5*, SC23-4278, because it is better to set up at the moment of the volume group creation or before the import is done on other nodes.

- d. Repeat this step for all the volume groups that you need to import on the server.
- e. Test the availability of the file systems by mounting them. To mount the file systems, run:

```
mount /udblogs/icmhadb
mount /udbdata/icmhadb
mount /home/db2fenc1
mount /home/db2inst1
```

If the file systems were mounted correctly, it is a good opportunity to verify and correct the owner, group, and permissions, because now there is a new mount point created. Refer to the end of 5.2.3, “File system setup and user IDs creation” on page 129 for details about how to do this.

3. Change volume groups to not activate at system restart.

Now, we need to change the auto varyon attribute to no. The reason for doing this is because HACMP must take care of the volume activation in order to avoid a node lock on any volume group when it starts. Suppose that you do not change this attribute, and both nodes (primary and backup) are powered off. Then you turn on the backup node first, and it activates the volume groups. If you turned on the primary server, it would not be possible to mount the volume groups, because the second one has already done it before. This is why you need to let the high availability software take care of that.

Following with our example, we change this attribute for the *vgdb2data* volume group.

You can accomplish this with smit:

- a. Enter:

```
smitty chvg
```

- b. Select **Change a Volume Group** and select the one that you want to change. See Figure 5-7 on page 147.



Change a Volume Group

Type or select values in entry fields.  
Press Enter AFTER making all desired changes.

|                                                                 | [Entry Fields] |   |
|-----------------------------------------------------------------|----------------|---|
| * VOLUME GROUP name                                             | vgdb2data      |   |
| * Activate volume group AUTOMATICALLY at system restart?        | no             | + |
| * A QUORUM of disks required to keep the volume group on-line ? | yes            | + |
| Convert this VG to Concurrent Capable?                          | no             | + |
| Change to big VG format?                                        | no             | + |
| LTG Size in kbytes                                              | 128            | + |
| Set hotspare characteristics                                    | n              | + |
| Set synchronization characteristics of stale partitions         | n              | + |

Figure 5-7 The chvg command smit panel

Alternatively, you can use the following command:

```
chvg -a 'n' -Q 'y' vgdb2data
```

Repeat this step for all the volume groups involved in the HACMP configuration (in our case, vgdb2bin, vgdb2data, and vgdb2logs).

### 5.2.6 DB2 and Content Manager instance creation on secondary node

In order for the Library Server to work properly on the backup node, it has to be prepared with the same data and configuration as the primary node. After following the configuration procedures in 5.2.4, “DB2 and Content Manager instance creation on primary node” on page 137, there are a few things that remain to be done on the secondary node to have the same environment available on both machines, and we explain these in this section.

The different parts that need to be configured are:

- ▶ DB2 instance
- ▶ Library Server database:
  - Database file systems
  - Transaction logs
- ▶ Content Manager user environments
- ▶ Access modules
- ▶ Library Server monitor

To do this, perform the following:

1. Create the DB2 instance on the secondary node.

In short, a DB2 instance consists of the instance owner user ID with its home directory and profile scripts, the directory structure under the sqllib directory, the DB2 binaries, and some additional information that tells the DB2 installation which instances should be present in that particular node.

In our scenario, the user ID is defined in both nodes with the same ID number, the file system for the home directory that holds the profile scripts and the sqllib directory is configured to be shared, and the DB2 binaries are at the same level in both machines. Also, the permissions for the local directories on which this file system is mounted were configured to be the same.

The only missing part is the information that tells DB2 that an instance is present in this second node also. This is the information under the /var directory. To create this information in the second node, use the standard DB2 utilities to create a new instance and then physically remove the created sqllib directory, leaving the information in /var intact. Of course, the information for the creation of this instance has to be the same as the information used for the creation of the original instance in the primary node.

To do this:

- a. Unmount the instance owner's home directory in the secondary node:

```
umount /home/db2inst1
```

- b. Verify that there is enough space in the file system that holds the /home/db2inst1 directory at this point. It should be enough with around 40 MB; you can verify this by looking at how much space is used on the sqllib directory of the instance that has been created in the primary node.

- c. Create the DB2 instance in the secondary node:

```
/usr/opt/db2_08_01/instance/db2icrt -u db2fenc1 db2inst1
```

- d. When the above program finishes, remove the files that were created in the home directory of the instance owner:

```
rm -Rf /home/db2inst1/*
```

Now everything should be ready to be able to use the DB2 instance on the secondary node after /home/db2inst1 is mounted.

2. Configure the Library Server database.

From a database point of view, both the tablespace and the transaction logs file systems are going to be shared between instances. Also, the database catalog is included with the database table spaces file system in the environment we defined. Thus, the only thing needed to be able to have the database available in the secondary node is to share the file systems, which, in this case, is already done.

### 3. Configure the Content Manager user environments.

We are going to share the file systems for the home directory of the DB2 instance owner and fenced ID, which will include their environment variables as well as the `profile.env` and `userprofile` files. Therefore, the only profile that needs to be updated is the profile of the Content Manager administrative user. In our case, it is `icmadmin`.

To do this, simply copy the `.profile` file located in the home directory of this user from the primary node to the secondary node.

### 4. Configure Content Manager generated access modules.

The file system that holds all the access modules is also shared between the two nodes, and thus nothing additional needs to be done.

### 5. Configure the Library Server monitor.

The Library Server monitor is a process that keeps track of the status of the Resource Managers associated with this Library Server. It must be running in order to allow failover and fallback between replicated Resource Managers.

For this process to run, it needs a configuration file called `DB2LS.properties` that is created in the `/usr/lpp/icm/config` directory when the Library Server database creation utility is run. You have to copy that file from the primary to the secondary node or simply edit the file adding the name of your Library Server database and the user ID used as the Content Manager administrator user ID. Example 5-10 shows a copy of the `DB2LS.properties` file in our environment.

*Example 5-10 DB2LS.properties file in both Library Server nodes*

---

```
LSDBNAME=ICMHADB
LSCMADM=icmadmin
```

---

## 5.2.7 HACMP topology configuration

In this section, we go through the setup steps to configure HACMP Version 4.5. We assume you have the Library Server already configured and running on the primary node *euclid*. You need to do this when logged in with the root user. The examples and configuration steps are based on the configuration that we used in our scenario.

**Important:** Do step 1 on both nodes and the following steps on *only one* of them. The configuration is later replicated to the other node. For our scenario, we choose to configure everything first from our primary node *euclid*.

To configure HACMP:

1. Set up the name resolution order and host definitions.

Before we define the adapters that HA will use, we need to fix the name resolution in both hosts for them to be able to resolve the names we are going to assign them.

- a. Update or create the `/etc/netsvc.conf` file to include the following syntax:

```
hosts=local,bind
```

This will force TCP/IP name resolution first to check the local `/etc/hosts` file for the name and then go to the name server. This is used to avoid name resolution issues between the nodes on the cluster if a network problem occurs, for example.

For HA, we need to define at least three IP addresses if we want to achieve IP address takeover (*IPAT*).

**Tip:** For a reference of service, boot, standby adapters, and IPAT, see Chapter 1, “Network Adapters,” in *Concepts and Facilities Guide*, SC23-4276. We recommend reading the chapter, because we are going to use this concept to configure the cluster.

We use only one service adapter, because we want HA on one service adapter only. The second one will be bound to the Content Manager Library Server. The third one will not be configured to achieve IPAT.

- b. Update the `/etc/hosts` file with the IP addresses/host names that we will be using for this configuration. Table 5-7 shows how our `/etc/hosts` file was configured. Note that we add the suffix `_svc` for service, `_boot` for boot, and `_stby` for the standby adapters. This is a good practice to avoid getting confused. Remember that you can add an alias for the service adapter.

Table 5-7 The `/etc/hosts` file configuration

| IP address   | Name                    | Resource group | IPAT |
|--------------|-------------------------|----------------|------|
| 9.30.130.66  | aristotle aristotle_svc | cm_dummy       | N    |
| 9.30.130.68  | aristotle_boot          | cm_dummy       | N    |
| 10.30.10.66  | aristotle_stby          | cm_dummy       | N    |
| 9.30.130.67  | euclid euclid_svc       | cm_rg          | Y    |
| 9.30.130.69  | euclid_boot             | cm_rg          | N    |
| 10.30.130.67 | euclid_stby             | cm_rg          | N    |

Example 5-11 shows an example of the hosts file that we defined.

*Example 5-11 Example of HACMP-related host name and address on /etc/hosts file*

---

|             |                         |
|-------------|-------------------------|
| 9.30.130.66 | aristotle aristotle_svc |
| 9.30.130.68 | aristotle_boot          |
| 10.30.10.66 | aristotle_stby          |
|             |                         |
| 9.30.130.67 | euclid euclid_svc       |
| 9.30.130.69 | euclid_boot             |
| 10.30.10.67 | euclid_stby             |

---

- c. Configure the IP address on the adapters on both nodes. The primary service adapter will be defined with the boot IP address, and the standby adapter to the standby IP address. Use the **smit chinnet** fast path, and select and define the adapters according to your configuration. After that, set the host name using **smit hostname**. This could be very important, because some applications might depend on that. For DB2 as an example, we remove that by changing the host name on the db2nodes.cfg file, but for other applications or environments, you cannot do that.

**Note:** You must assign different subnets between the service/boot and standby adapters; also the netmask for *all* adapters must be the same to avoid communication problems between standby adapters after an adapter swap. The communication problem occurs when the standby adapter assumes its original address but retains the netmask of the takeover address.

Continuing with our example, the *euclid\_svc* address will be the one that the Library Server clients will connect to. On the other service adapter, we will define a “dummy” resource group in order to simulate another application that could be running when the backup node is waiting for the takeover to happen. We discuss that later.

- d. Update the root \$HOME/.rhosts file and make sure that the entries for service, boot, and standby adapters are included for all the participating nodes on the cluster, as shown in Example 5-12.

*Example 5-12 Example of the .rhosts file*

---

|             |      |
|-------------|------|
| 9.30.130.66 | root |
| 9.30.130.67 | root |
| 9.30.130.68 | root |
| 9.30.130.69 | root |
| 10.30.10.66 | root |
| 10.30.10.67 | root |

---

**Note:** Do not mix configuration steps between different nodes because this could lead to an inconsistent situation. Remember that step 1 is the *only one* that you need to run on all the nodes that are part of the cluster.

2. Set the cluster ID.

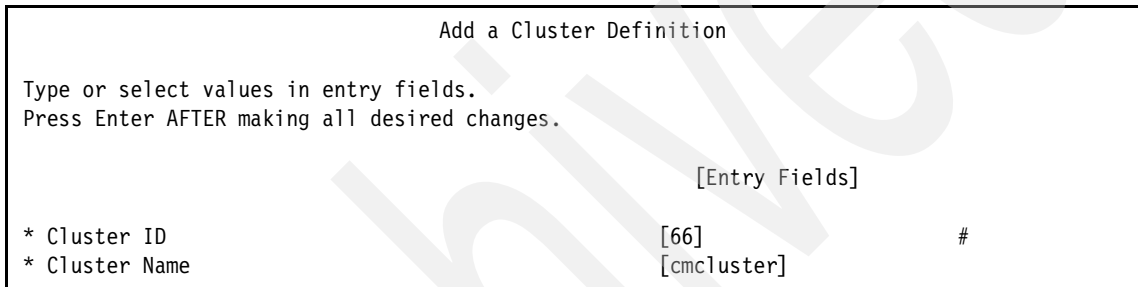
You can accomplish this with smitty:

a. Enter:

```
smitty hacmp
```

b. Select **Cluster Configuration** → **Cluster Topology** → **Configure Cluster** → **Add a Cluster Definition**.

c. Enter the appropriate values, as shown in Figure 5-8.



Add a Cluster Definition

Type or select values in entry fields.  
Press Enter AFTER making all desired changes.

[Entry Fields]

\* Cluster ID [66] #  
\* Cluster Name [cmcluster]

Figure 5-8 Add a Cluster Definition smit panel

Alternatively, you can use the command line:

```
/usr/sbin/cluster/utilities/claddclstr -i'1' -n'cmcluster'
```

**Note:** It is very important that the cluster ID chosen is unique, because you cannot have two HACMP systems running with the same cluster ID on the same network. Ask your HACMP/AIX administrator whether there is another machine running HACMP software.

3. Add nodes.

You can accomplish this with smitty:

a. Enter:

```
smitty hacmp
```

b. Select **Cluster Configuration** → **Cluster Topology** → **ConfigureNodes** → **Add Cluster Nodes**.

c. Enter the appropriate values, as shown in Figure 5-9 on page 153.

| Add Cluster Nodes                                                                       |                                      |
|-----------------------------------------------------------------------------------------|--------------------------------------|
| Type or select values in entry fields.<br>Press Enter AFTER making all desired changes. |                                      |
| * Node Names                                                                            | [Entry Fields]<br>[euclid aristotle] |

Figure 5-9 Add Cluster Nodes smit panel

Alternatively, you can use the command:

```
/usr/sbin/cluster/utilities/clnodename -a 'aristotle euclid'
```

4. Add the networks.

You can accomplish this with smitty:

a. Enter:

```
smitty hacmp
```

b. Select **Cluster Configuration** → **Cluster Topology** → **Configure Networks** → **Configure IP-based Networks** → **Add a Network**.

c. Enter the appropriate values, as shown in Figure 5-10.

| Add an IP-based Network                                                                 |                              |
|-----------------------------------------------------------------------------------------|------------------------------|
| Type or select values in entry fields.<br>Press Enter AFTER making all desired changes. |                              |
| * Network Name                                                                          | [Entry Fields]<br>[cm_ether] |
| * Network Attribute                                                                     | public +                     |
| Network Type                                                                            | [ether] +                    |
| Subnet(s)                                                                               | [10.30.10.0/25 9.30.13> +    |

Figure 5-10 Add an IP-based Network smit panel

Alternatively, you can use the command:

```
/usr/sbin/cluster/utilities/clmodnetwork -a -n'cm_ether' -t'public' -i\
'ether'
```

Do not worry about errors saying that there is no adapter defined. We define them later.

5. Add non-P network adapters.

You can accomplish this with smitty:

a. Enter:

```
smitty hacmp
```

b. Select **Cluster Configuration** → **Cluster Topology** → **Configure Networks** → **Configure Non IP-based Networks** → **Add a Network**.

c. Enter the appropriate values, as shown in Figure 5-11.

Add a Non IP-based Network

Type or select values in entry fields.  
Press Enter AFTER making all desired changes.

\* Network Name

\* Network Type

[Entry Fields]

[cm\_serial]

[rs232]

+

Figure 5-11 Add a Non IP-based Network smit panel

Alternatively, you can use the command:

```
/usr/sbin/cluster/utilities/clmodnetwork -a -n'cm_serial' -t serial\ -i
'rs232'
```

We highly recommend that you configure your cluster with a serial interface, because you are avoiding TCP/IP to be the single point of failure. For information about configuring the devices, see Chapter 2, in *Installation Guide for HACMP V4.5, SC23-4278*.

6. Add IP-based adapters.

Now, we add the three adapters (service, standby, and boot) to the Topology definition.

You can accomplish this with smit:

a. Enter:

```
smit hacmp
```

b. Select **Cluster Configuration** → **Cluster Topology** → **Configure Adapters** → **Adapters on IP-based network** → **Add an Adapter (cm\_ether)**.

c. Enter the appropriate values for the service adapter configuration, as shown in Figure 5-12 on page 155.

d. Enter the appropriate values for the boot adapter configuration, as shown in Figure 5-13 on page 156.



- e. Enter the appropriate values for the standby adapter configuration, as shown in Figure 5-14 on page 156.

Add an IP-based Adapter

Type or select values in entry fields.  
Press Enter AFTER making all desired changes.

|                          | [Entry Fields]      |   |
|--------------------------|---------------------|---|
| * Adapter IP Label       | [euclid]            | + |
| Network Type             | ether               |   |
| Network Name             | cm_ether            |   |
| * Adapter Function       | [service]           | + |
| Adapter IP address       | [9.30.130.67]       |   |
| Adapter Hardware Address | [00.02.55.33.1e.f8] |   |
| Node Name                | [euclid]            | + |
| Netmask                  | [255.255.255.128]   | + |

Figure 5-12 Add an IP-based adapter, service adapter configuration

Because we will use hardware address swapping, we need to define an alternate hardware address for the *service* adapter. This is because if a takeover takes place, we want to keep the same hardware address along with the IP address to avoid loss of communications with the clients connected.

To specify an alternate hardware address for an Ethernet interface, begin by using the first five pairs of alphanumeric characters as they appear in the current hardware address. Then substitute a different value for the last pair of characters. Use characters that do not occur on any other adapter on the physical network.

For our example, the original address is 00.02.55.33.1e.f3, and the new address is 00.02.55.33.1e.f8.

**Note:** Check to make sure there is no duplicated hardware address configured within the network. To check the hardware address, use the command `netstat -i | grep link`.

| Add an IP-based Adapter                                                                 |                   |   |
|-----------------------------------------------------------------------------------------|-------------------|---|
| Type or select values in entry fields.<br>Press Enter AFTER making all desired changes. |                   |   |
|                                                                                         | [Entry Fields]    |   |
| * Adapter IP Label                                                                      | [euclid_boot]     | + |
| Network Type                                                                            | ether             |   |
| Network Name                                                                            | cm_ether          |   |
| * Adapter Function                                                                      | [service]         | + |
| Adapter IP address                                                                      | [9.30.130.69]     |   |
| Adapter Hardware Address                                                                | []                |   |
| Node Name                                                                               | [euclid]          | + |
| Netmask                                                                                 | [255.255.255.128] | + |

Figure 5-13 Add an IP-based Adapter, boot adapter configuration

| Add an IP-based Adapter                                                                 |                   |   |
|-----------------------------------------------------------------------------------------|-------------------|---|
| Type or select values in entry fields.<br>Press Enter AFTER making all desired changes. |                   |   |
|                                                                                         | [Entry Fields]    |   |
| * Adapter IP Label                                                                      | [euclid_stby]     | + |
| Network Type                                                                            | ether             |   |
| Network Name                                                                            | cm_ether          |   |
| * Adapter Function                                                                      | [service]         | + |
| Adapter IP address                                                                      | [10.30.10.67]     |   |
| Adapter Hardware Address                                                                | []                |   |
| Node Name                                                                               | [euclid]          | + |
| Netmask                                                                                 | [255.255.255.128] | + |

Figure 5-14 Add an IP-based Adapter, standby adapter configuration

Alternatively, you can use the following commands:

```
/usr/sbin/cluster/utilities/claddnode -a'euclid' : 'ether' : 'cm_ether' : \
: 'service' : '9.30.130.67' : -n'euclid' -m'255.255.255.128'
/usr/sbin/cluster/utilities/claddnode -a'euclid_boot' : 'ether'\
: 'cm_ether' : : 'boot' : '9.30.130.69' : -n'euclid' -m'255.255.255.128'
/usr/sbin/cluster/utilities/claddnode -a'euclid_stby' : 'ether'\
: 'cm_ether' : : 'standby' : '10.30.10.67' : -n'euclid' -m'255.255.255.128'
```

Configure the other nodes adapters as follows:

```
/usr/sbin/cluster/utilities/claddnode -a'aristotle' : 'ether' : 'cm_ether'\
: : 'service' : '9.30.130.66' : -n'aristotle' -m'255.255.255.128'
/usr/sbin/cluster/utilities/claddnode -a'aristotle_boot' : 'ether'\
: 'cm_ether' : : 'boot' : '9.30.130.68' : -n'aristotle' -m'255.255.255.128'
/usr/sbin/cluster/utilities/claddnode -a'aristotle_stby' : 'ether'\
: 'cm_ether' : : 'standby' : '10.30.10.66' : -n'aristotle' -m'255.255.255.128'
```

7. Add the non IP-based adapter.

You can accomplish this with smit:

a. Enter:

```
smit hacmp
```

b. Select **Cluster Configuration** → **Cluster Topology** → **Configure Adapters** → **Adapters on Non IP-based network** → **Add an Adapter (cm\_serial)**.

c. Enter the appropriate values for euclid, as shown in Figure 5-15.

d. Do the same for aristotle, as shown in Figure 5-16 on page 158.

| Add a Non IP-based Adapter                                                              |                 |
|-----------------------------------------------------------------------------------------|-----------------|
| Type or select values in entry fields.<br>Press Enter AFTER making all desired changes. |                 |
|                                                                                         | [Entry Fields]  |
| * Adapter Label                                                                         | [euclid_serial] |
| Network Type                                                                            | rs232           |
| Network Name                                                                            | cm_serial       |
| * Device Name                                                                           | [/dev/tty1]     |
| * Node Name                                                                             | [euclid] +      |

Figure 5-15 Add a Non IP-based Adapter smit panel

Add a Non IP-based Adapter

Type or select values in entry fields.  
Press Enter AFTER making all desired changes.

|                 |                    |   |
|-----------------|--------------------|---|
|                 | [Entry Fields]     |   |
| * Adapter Label | [aristotle_serial] |   |
| Network Type    | rs232              |   |
| Network Name    | cm_serial          |   |
| * Device Name   | [/dev/tty1]        |   |
| * Node Name     | [aristotle]        | + |

Figure 5-16 Add a Non IP-based Adapter smit panel

Alternatively, you can use the following commands:

```
/usr/sbin/cluster/utilities/claddnode -a'aristotle_serial' : 'rs232' \
:'cm_serial' :serial:service:'/dev/tty1' -n'aristotle' \
/usr/sbin/cluster/utilities/claddnode -a'euclid_serial' : 'rs232' \
:'cm_serial' :serial:service:'/dev/tty1' -n'euclid'
```

8. Show the topology configuration.

You can accomplish this with smitty:

- a. Enter:  
# smitty hacmp
- b. Select **Cluster Configuration** → **Cluster Topology** → **Show Cluster Topology** → **Show Cluster Topology**. See Example 5-13.

Example 5-13 Show Topology screen

| COMMAND STATUS                                                       |             |            |
|----------------------------------------------------------------------|-------------|------------|
| Command: OK                                                          | stdout: yes | stderr: no |
| Before command completion, additional instructions may appear below. |             |            |
| [TOP]                                                                |             |            |
| Cluster Description of Cluster cmcluster                             |             |            |
| Cluster ID: 66                                                       |             |            |
| There were 2 networks defined : cm_ether, cm_serial                  |             |            |
| There are 2 nodes in this cluster.                                   |             |            |
| NODE aristotle:                                                      |             |            |
| This node has 2 service interface(s):                                |             |            |
| Service Interface aristotle_svc:                                     |             |            |
| IP address: 9.30.130.66                                              |             |            |

Hardware Address:  
Network: cm\_ether  
[MORE...81]

|             |            |           |            |
|-------------|------------|-----------|------------|
| F1=Help     | F2=Refresh | F3=Cancel | F6=Command |
| F8=Image    | F9=Shell   | F10=Exit  | /=Find     |
| n=Find Next |            |           |            |

---

This is a good moment to review the configuration that you did in the previous steps and check if something is not as expected. It is better to change it at this moment so that we can synchronize the data between the nodes, which is the next step.

9. Synchronize the cluster topology.

You can accomplish this with smitty:

a. Enter:

```
smitty hacmp
```

b. Select **Cluster Configuration** → **Cluster Topology** → **Synchronize Cluster Topology**. See Figure 5-17 on page 160.

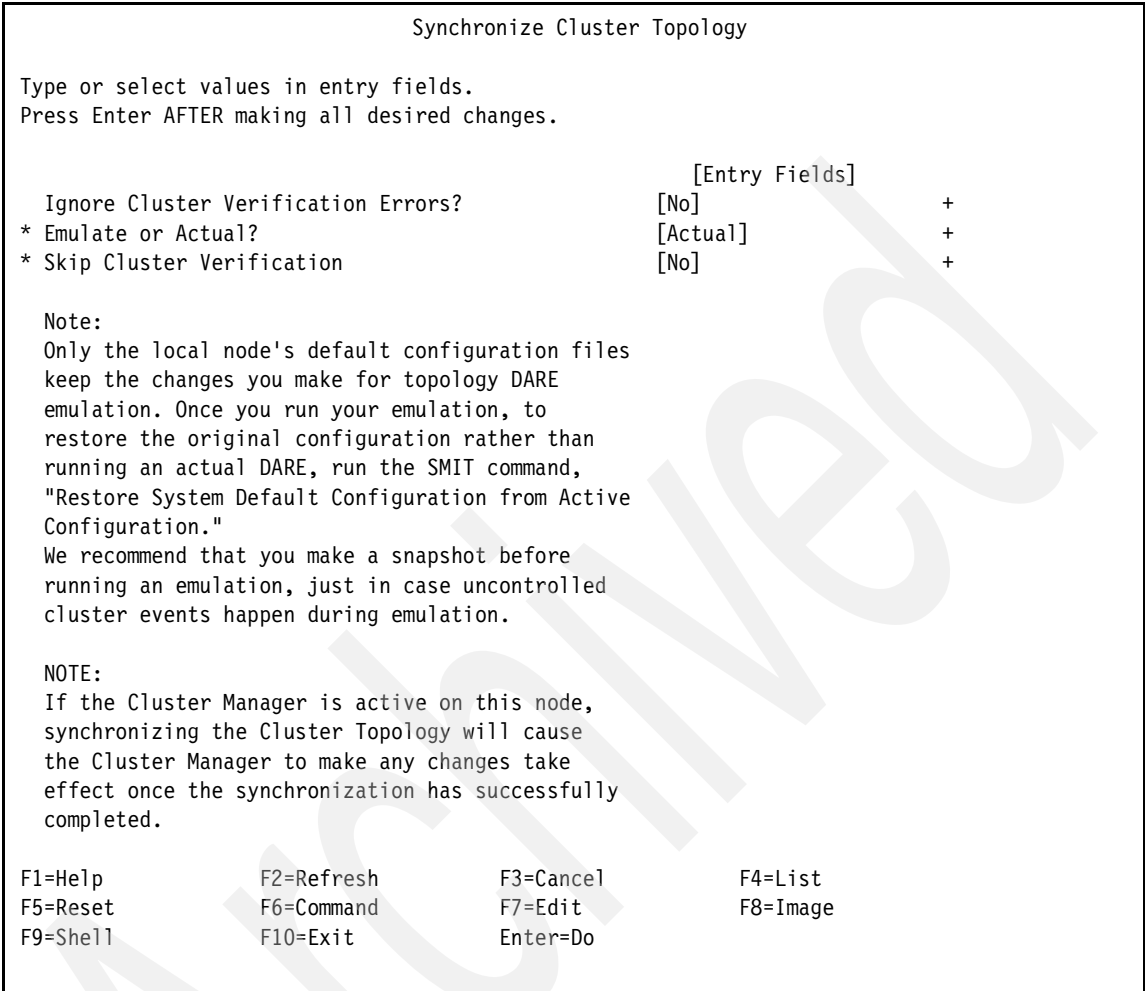


Figure 5-17 Synchronize Cluster Topology smit panel

Take a look to see if any errors appear. If the synchronization process finishes with a status other than “OK”, the entire process will not be done. Check and correct any errors.

### 5.2.8 Defining resource groups

With the topology distributed and synchronized to the nodes that are part of the cluster, we are ready to set the resource group of the cluster.

In our scenario, the resource group consists of all the components needed for the Library Server to run. These components are, in summary, the IP address of the Library Server, the shared file systems, and the different processes that need to be started, which include the DB2 database manager for the instance we created and the Library Server monitor.

The resource group configuration will define how and where the Library Server will run in a case of failure (failover) or takeover. As we mentioned earlier, we are going to define two resource groups. One is called *cm\_rg* and will take care of the Library Server. The relationship between the participating nodes is defined as *cascading without fallback*. This means that if the primary nodes fails, the backup will bring the Library Server online. The resource group will be active on the primary node until a resource group migration or a cluster stop occurs. We did that because we can have more control of the application and it is usually easier to administer this way.

We also define another resource group on the cluster that runs only on *aristotle*. This is intended to reflect the fact that you can have another application running on the backup node. This resource group is called *cm\_dummy* and contains only the service IP address of *aristotle* and no file systems or volume groups, and it will *only* belong to *aristotle*. No takeover will be done with this resource group. You could, however, define the relationships to fall back this resource group to the other node or put it offline if a Library Server takeover occurs.

In order to run the Library Server under the scope of HACMP, an application server cluster resource will be created that associates a user-defined name with the names of user-provided written scripts to start and stop the Library Server. For more information, refer to Chapter 10 of the *Installation Guide for HACMP V4.5*, SC23-4278.

Typically, the flow of the scripts is as follows:

- ▶ Start script:
  - a. Clean up the scripts.
  - b. Change the file or file configurations regarding the node changes.
  - c. Run the application startup scripts.
  - d. Check for the status.
- ▶ Stop script:
  - a. Run the application shutdown scripts.
  - b. Check for the application stop status.

Our start and stop scripts along with their explanation are detailed in A.1, “HACMP Library Server startup and shutdown scripts” on page 246. In our sample environment, these scripts are stored in the files *stopcm\_as.sh* and *startcm\_as.sh*.

We continue with the HACMP configuration process. We are going to define a resource group, an application server, and after synchronizing the data, we will be ready to test if the application runs as expected. Do the following:

1. Define resource groups.

You can accomplish this with smitty:

- a. Enter:

```
smitty hacmp
```

- b. Select **Cluster Configuration** → **Cluster Resources** → **Define a Resource Group** → **Add a Resource Group**.

- c. Enter the appropriate values, as shown in Figure 5-18.

Add a Resource Group

Type or select values in entry fields.  
Press Enter AFTER making all desired changes.

|                            | [Entry Fields]     |   |
|----------------------------|--------------------|---|
| * Resource Group Name      | [cm_rg]            |   |
| * Node Relationship        | cascading          | + |
| * Participating Node Names | [euclid aristotle] | + |

Figure 5-18 Add a Resource Group smit panel

Alternatively, you can use the following command:

```
/usr/sbin/cluster/utilities/claddgrp -g 'cm_rg' -r 'cascading' -n\
'euclid aristotle'
```

**Important:** The order in the node names field is very important because this defines which node has the highest priority for the resources group. Pay special attention that you put first the node where the Library Server will be mainly running.

2. Add a resource group.

You can accomplish this with smitty:

- a. Enter:

```
smitty hacmp
```

- b. Select **Cluster Configuration** → **Cluster Resources** → **Define a Resource Group** → **Add a Resource Group**.

- c. Enter the appropriate values, as shown in Figure 5-19 on page 163.



Add a Resource Group

Type or select values in entry fields.  
Press Enter AFTER making all desired changes.

|                            | [Entry Fields] |   |
|----------------------------|----------------|---|
| * Resource Group Name      | [cm_dummy]     |   |
| * Node Relationship        | cascading      | + |
| * Participating Node Names | [aristotle]    | + |

Figure 5-19 Add a Resource Group smit panel

Alternatively, you can use the following command:

```
/usr/sbin/cluster/utilities/claddgrp -g 'cm_dummy' -r 'cascading' -n 'aristotle'
```

3. Define the application servers.

You can accomplish this with smitty:

- a. Enter:  
# smitty hacmp
- b. Select **Cluster Configuration** → **Cluster Resources** → **Define Application Servers** → **Add an Application Server**.
- c. Enter the appropriate values, as shown in Figure 5-20.

Add an Application Server

Type or select values in entry fields.  
Press Enter AFTER making all desired changes.

|                | [Entry Fields]                 |
|----------------|--------------------------------|
| * Server Name  | [cm_as]                        |
| * Start Script | [/home/icmadmin/startcm_as.sh] |
| * Stop Script  | [/home/icmadmin/stopcm_as.sh]  |

Figure 5-20 Add an Application Server smit panel

Alternatively, you can use the following command:

```
/usr/sbin/cluster/utilities/claddsrv -s'cm_as' -b'/home/icmadmin/startcm_as.sh' -e'/home/icmadmin/stopcm_as.sh'
```

Place the file names of your Library Server start and stop scripts in the appropriate fields.

4. Change the resources/attributes for the resource group.

Here you need to define the node relationship, participating nodes, and the different resources involved on the resource group.

You can accomplish this with smitty:

a. Enter:

```
smitty hacmp
```

b. Select **Cluster Configuration** → **Cluster Resources** → **Change/Show Resources/Attributes for a Resource Group (cm\_rg)**.

c. Enter the appropriate values, as shown in Figure 5-21.

Change/Show Resources/Attributes for a Resource Group

Type or select values in entry fields.  
Press Enter AFTER making all desired changes.

|                                    | [Entry Fields]                 |   |
|------------------------------------|--------------------------------|---|
| Resource Group Name                | cm_rg                          |   |
| Node Relationship                  | cascading                      |   |
| Participating Node Names           | euclid aristotle               |   |
| Service IP label                   | [euclid_svc]                   | + |
| Filesystems (default is All)       | [/home/db2inst1 /home/db2fenc1 |   |
| /udbdata/icmhadb /udblogs/icmhadb] |                                | + |
| Volume Groups                      | [vgdb2logs vgdb2data vgdb2bin] | + |
| Application Servers                | [cm_as]                        | + |
| Cascading Without Fallback Enabled | true                           | + |

Figure 5-21 Change/Show Resources/attributes for a Resource Group smit panel for cm\_rg

d. Do the same for the other resource group, as shown in Figure 5-22 on page 165.

Change/Show Resources/Attributes for a Resource Group

Type or select values in entry fields.  
Press Enter AFTER making all desired changes.

|                          | [Entry Fields]  |   |
|--------------------------|-----------------|---|
| Resource Group Name      | cm_dummy        |   |
| Node Relationship        | cascading       |   |
| Participating Node Names | aristotle       |   |
| Service IP label         | [aristotle_svc] | + |

Figure 5-22 Change/Show Resources/Attributes for a Resource Group smit panel for cm\_dummy

**Note:** Only the changed options are shown here.

5. Synchronize the cluster resources.  
You can accomplish this with smitty:
  - a. Enter:  
# smitty hacmp
  - b. Select **Cluster Configuration** → **Cluster Resources** → **Synchronize Cluster Resources**.
  - c. Enter the appropriate values, as shown in Figure 5-23 on page 166.

Synchronize Cluster Resources

Type or select values in entry fields.  
Press Enter AFTER making all desired changes.

|                                     | [Entry Fields] |   |
|-------------------------------------|----------------|---|
| Ignore Cluster Verification Errors? | [No]           | + |
| Un/Configure Cluster Resources?     | [Yes]          | + |
| * Emulate or Actual?                | [Actual]       | + |
| * Skip Cluster Verification         | [No]           | + |

Note:  
Only the local node's default configuration files keep the changes you make for resource DARE emulation. Once you run your emulation, to restore the original configuration rather than running an actual DARE, run the SMIT command, "Restore System Default Configuration from Active Configuration."  
We recommend that you make a snapshot before running an emulation, just in case uncontrolled cluster events happen during emulation.

Figure 5-23 Synchronize Cluster Resources smit panel

Alternatively, you can use the following command:

```
/usr/sbin/cluster/utilities/cldare -r
```

Pay special attention to the result of this operation, because if the synchronization finds any error, the configuration will not be passed to the other nodes on the cluster, and you will need to correct the errors and try this process again. After the synchronization process finishes, this is a good time to test if the different components that you set up are working as expected.

## 5.2.9 HACMP post-configuration procedures

To start the cluster services on any of the nodes, issue the following command:

```
smit clstart
```

This starts the cluster daemon (clstrmgr) and brings up all the resources involved (IP address, file systems, and the Library Server itself).

To stop the cluster services, issue the following command:

```
smit clstop
```

With the configuration we performed, the Library Server resource group will be running on the primary node during normal operation. You can bring the resource group online or offline manually. You will need to do this when a takeover takes place and you want to bring the Library Server back to the primary node, or if you want to shut down the primary node for maintenance.

There are two ways you can move the Library Server from one node to the other. One way is to bring the resource group offline from the original node where it is running and bring it online on the desired node where you want it running. Another way to do the same thing is to migrate the resource from one node to the other. You can go to the menu for cluster resource group management by using the following command:

```
smit cl_resgrp
```

If you want to bring a resource group offline, select the corresponding menu item (keep in mind that all the connections to the Content Manager system will be severed), the Library Server will be stopped, the file systems unmounted, and the IP address associated to that resource group will be unconfigured. See Figure 5-24.

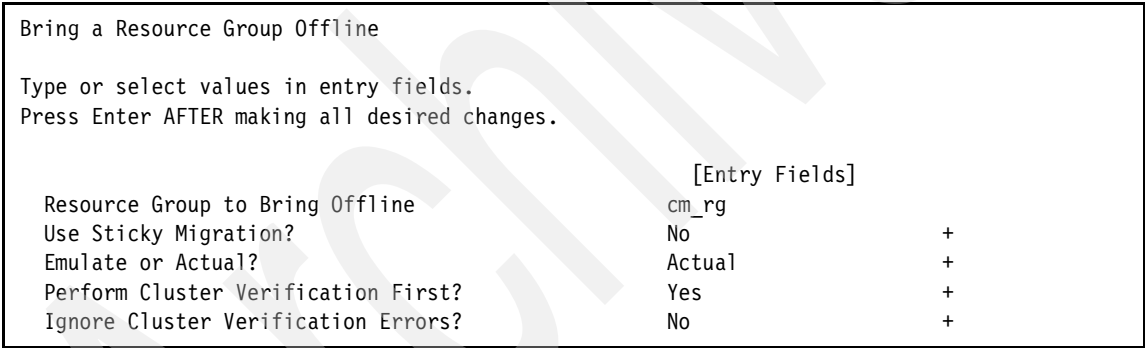


Figure 5-24 Bring a Resource Group Offline smit panel

Also, if you want to know where the resource groups are running and their status, run:

```
/usr/sbin/cluster/utilities/clfindres
```

This command tells you the location of the resources and the node where it is active.

**Important:** We highly recommend that you read *Administration Guide for HACMP V4.5*, SC23-4279, if you are going to administer the HACMP installation.

## 5.3 Resource Manager replication

As mentioned in 5.1, “Introduction” on page 122, high availability for the Resource Manager component in our environment is achieved by using Resource Manager replication. The idea behind this concept is that every object in the primary Resource Manager is periodically replicated to one or more separate Resource Managers. In the case of a Resource Manager failure, the Content Manager system reroutes the resource object-related requests to any other Resource Manager that has a replica of that particular object. This Content Manager feature is further explained in 4.2.4, “High availability example 2: Clustering and replication” on page 110.

In our scenario, we have only two Resource Managers. The one running on harm01vm is our primary Resource Manager and is assigned as the default Resource Manager for every user and item type defined in the system. If no failures occur with this Resource Manager, this would be the only one ever used to service user requests.

The Resource Manager running on harm02vm is acting as a backup for harm01vm. Every resource item that is stored in harm01vm is replicated to harm02vm, and, in case of a failure in harm01vm, the Library Server switches operation to the second Resource Manager, making harm02vm services all resource item requests from that point on. When the primary Resource Manager (harm01vm) is up again, the Library Server automatically switches back to this Resource Manager (harm01vm), and all operation goes back to normal.

### 5.3.1 Initial parameter definitions

For the purpose of our tests, two collections are created in harm01vm: one that stores objects forever on disk and one that stores objects forever on Tivoli Storage Manager. Each of these collections is defined to replicate against a similar collection defined in harm02vm. Both Resource Managers have a similar configuration consisting of a single local file system volume and a single Tivoli Storage Manager volume connecting to cm06. Item types created for the fail-over tests use either one of these collections.

Table 5-8 on page 169, Table 5-9 on page 169, and Table 5-10 on page 169 list the values for the most relevant settings that are used in our installation. We use default values for all other parameters.

Table 5-8 Settings for the primary Resource Manager

| Name                             | Value    | Comments |
|----------------------------------|----------|----------|
| Host name                        | harm01vm |          |
| RM database name                 | HARMDB01 |          |
| Tivoli Storage Manager node name | HARM01   |          |

Table 5-9 Settings for the secondary Resource Manager

| Name                             | Value    | Comments |
|----------------------------------|----------|----------|
| host name                        | harm02vm |          |
| RM database name                 | HARMDB02 |          |
| Tivoli Storage Manager node name | HARM02   |          |

Table 5-10 Settings common to both Resource Managers

| Name                                    | Value              | Comments                                                                              |
|-----------------------------------------|--------------------|---------------------------------------------------------------------------------------|
| Library Server host name                | euclid.svl.ibm.com |                                                                                       |
| Database port number                    | 60000              |                                                                                       |
| LS database name                        | ICMHADB            |                                                                                       |
| LS administration user ID               | icmadmin           |                                                                                       |
| Tivoli Storage Manager server host name | cm06.svl.ibm.com   |                                                                                       |
| Tivoli Storage Manager management class | HACCLASS           | The same Tivoli Storage Manager management class is shared by both Resource Managers. |

We recommend that you create a similar table with your own Resource Manager information before performing any installation or configuration steps.

### 5.3.2 Resource Manager installation and configuration

The steps we go through in this section assume that you are setting up a new Content Manager system, including the first time installation of the Resource Managers. This was the case for our test scenario. If you have a Content Manager system, and your primary Resource Manager is already up and

running, you can use the following steps to install only your backup Resource Manager.

Before continuing to 5.3.3, “Cross-referencing server definitions” on page 171, both the primary and backup Resource Managers have to be installed and configured to work properly as part of the Content Manager system. This includes the installation of all the required software components, the creation of the Resource Manager database, and the deployment of the Resource Manager Web application in WebSphere Application Server.

To have everything ready for the configuration of each Resource Manager, you should first install all the required software. For our environment, we installed the following software on harm01vm and harm02vm:

- ▶ Microsoft Windows 2000 Server with Service Pack 3
- ▶ DB2 UDB ESE Version 8.1 with Fix Pack 3
- ▶ WebSphere Application Server Version 5.0 with Fix Pack 3
- ▶ Tivoli Storage Manager client Version 5.2
- ▶ DB2 Content Manager for Multiplatforms Version 8.2 with Fix Pack 2

All the software should be installed using the default options. For the Content Manager installation, choose to install only the Resource Manager component. Use the values from Table 5-8 on page 169, Table 5-9 on page 169, and Table 5-10 on page 169 to create the Resource Manager databases and deploy the Resource Manager application in both servers. Do not install or create any Library Server databases on these machines.

If you encounter any problem with the automatic configuration procedures, use the manual programs explained in Chapter 40, “Using Content Manager after-install programs and procedures,” in *Planning and Installing Your Content Management System*, GC27-1332.

Do not forget to complete all of the following steps to make sure all the components of both Resource Managers are running:

1. Create the Resource Manager database.
2. Deploy the Resource Manager application in WebSphere Application Server.
3. Create the corresponding Resource Manager services (replicator, migrator, purger, and stager).
4. Add both Resource Managers to the Library Server configuration.
5. Configure access to the Tivoli Storage Manager server if appropriate.



Before going on to the next step, you need to make sure that both Resource Managers are working. You can do a fairly complete test by doing the following for each Resource Manager:

1. Perform the necessary SMS definitions to create a new Resource Manager collection that stores objects forever in Tivoli Storage Manager.
2. Define a new item type that stores resource parts on this collection.
3. Store a few objects into this item type using one of the Content Manager clients.
4. Verify that you are able to access them by viewing some of them.
5. Query the Tivoli Storage Manager server for files stored by the node name corresponding to this Resource Manager and take note of their names.
6. Delete the imported objects using one of the Content Manager clients.
7. Let a few minutes pass by to let the migrator run in the background.
8. Force an inventory expiration in Tivoli Storage Manager.
9. Verify that the files found in step 5 no longer exist in Tivoli Storage Manager.

**Tip:** If you are using Tivoli Storage Manager and receive a message such as the following one in the migrator or replicator log files, you can fix it by adding the complete path to the specified file in the PATH system environment variable and restarting the RM services:

```
ICMRM:ERROR 2003-11-11 10:44:48,031 [main] - Can't load library:
/C:/Program%20Files/WebSphere/AppServer/installedApps/HARM02VM/icrm.ear/icrm.war/bin/DMAeJAdsmApi.dll - <clinit>(DMAeJAdsmApi.java:2137)
```

### 5.3.3 Cross-referencing server definitions

This is the first step specific to the configuration of Resource Manager replication. Here, we create and modify all the necessary server definitions for the different components of the Content Manager system to be able to talk to each other:

1. Update the server definitions for the primary Resource Manager.
  - a. Using the Content Manager System Administration Client, log in to the Content Manager system.
  - b. Navigate to the Server Definitions node of the first Resource Manager on the left panel. For our scenario, the path to select would be **ICMHADB → Resource Managers → HARMDB01 → Server Definitions**.

Here, you will probably see three server definitions: one for the Library Server, one for the primary Resource Manager, and one for the Tivoli Storage Manager server.

- c. Right-click the name of the primary Resource Manager database on the right panel and select **Properties**. Verify that the Resource Manager definition for the primary Resource Manager is correct. Pay special attention to the Protocol and Port number fields. These should match the ones specified during the Resource Manager Web application deployment. Figure 5-25 shows the settings for the primary server in our test environment.

Server Definition Properties - HARMDB01

Name: \* HARMDB01

Server type: Resource Manager

Hostname: \* localhost

Platform: \* Windows

User ID: \* rmadmin

Password: \* \*\*\*\*\*

Protocol: http

Port number: \* 80

Schema: \* rmadmin

Path: \* /icmrm/ICMResourceManager

OK Cancel Apply Help

Figure 5-25 Server Definition for HARMDB01

- d. In the same location within the System Administration Client, create a new server definition for the standby Resource Manager by right-clicking the **Server Definitions** node on the left panel and selecting **New**.
- e. Fill in all the required information. Figure 5-26 on page 173 shows the settings in our test environment.

**New Server Definition**

Name: \* HARMDB02

Server type: Resource Manager

Hostname: \* harm02vm.svl.ibm.com

Platform: \* Windows

User ID: \* radmin

Password: \* \*\*\*\*\*

Protocol: http

Port number: \* 80

Schema: \* radmin

Path: \* /icmrm/ICMResourceManager

OK Cancel Apply Help

Figure 5-26 Server Definition for HARMDB02

After completing these steps, the services assigned to the primary Resource Manager should be able to contact the standby Resource Manager during replication.

2. Update the server definitions for the secondary Resource Manager.
  - a. Repeat the same steps for the server definitions within the secondary Resource Manager. Use the System Administration Client to navigate to the Server Definitions node inside the secondary Resource Manager, which in our case would be **ICMHADB → Resource Managers → HARMDB02 → Server Definitions**.
  - b. Verify the Port number and Protocol definitions in the secondary Resource Manager definition.
  - c. Create a new server definition for the primary Resource Manager.

### 5.3.4 Enable collections for replication

The next step in our sample environment is to enable replication for each collection that exists on the primary Resource Manager. There is no restriction on how the collections have to be set up in the secondary Resource Manager for a particular primary collection. You can define a single collection in the secondary Resource Manager to hold the replicas of all the collections in the other Resource Managers, you can create one collection on the secondary

Resource Manager for each collection on the primary Resource Manager, or you can even replicate each collection on the primary Resource Manager to a collection in a different secondary Resource Manager.

In our case, we defined two collections in the primary server: one for local disk storage and one for Tivoli Storage Manager storage. We also created similar collections on the secondary Resource Manager, and then replicated each collection of the primary Resource Manager to the corresponding collection in the secondary Resource Manager. This way, we achieve our idea of a replicated standby Resource Manager for high availability.

The SMS configuration for the collections that are defined in both of our Resource Managers is shown in Figure 5-27 and Figure 5-28.

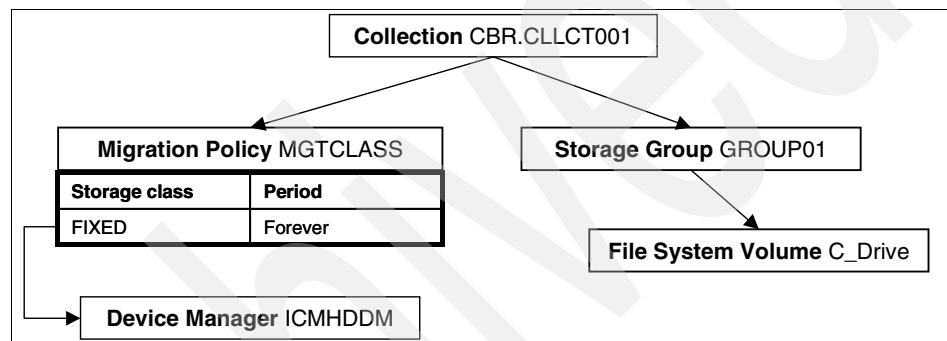


Figure 5-27 CM workstation collection configuration for local disk storage

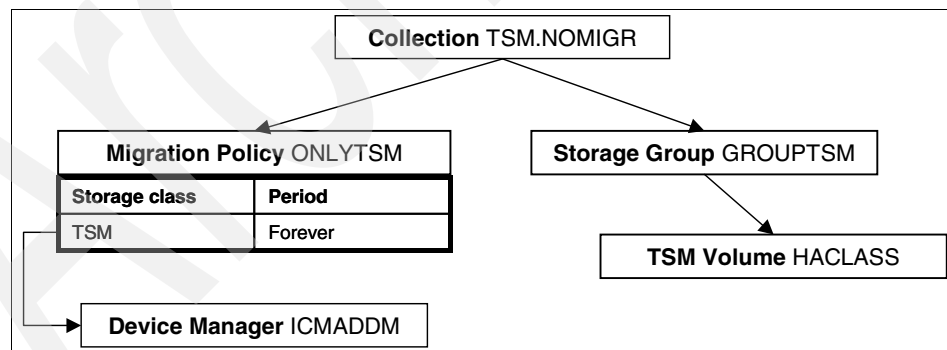


Figure 5-28 CM workstation collection for Tivoli Storage Manager only storage

To enable replication of objects for high availability, you need to configure each workstation collection in the primary Resource Manager to replicate to a collection in the standby Resource Manager. In our case, we need to do this for both collections, CBR.CLLCT001 and TSM.NOMIGR.

We do this using the Content Manager System Administration Client:

1. Navigate to the Workstation Collections node within the primary Resource Manager by selecting **ICMHADB** → **Resource Managers** → **HARMDB01** → **Workstation Collections**.
2. On the right panel, double-click the name of the collection.
3. On the Workstation Collection Properties panel, click **Add**.
4. Select the secondary Resource Manager in the Resource Manager pull-down, select the remote collection name in the Collection pull-down, and click **OK**.

Figure 5-29 shows the resulting configuration for the local disk storage collection in our sample scenario.

5. Click **OK** to finish the replication definition.

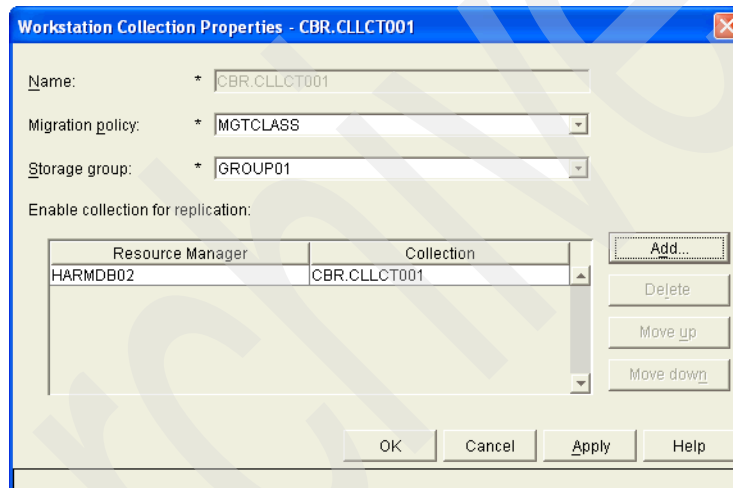


Figure 5-29 CBR.CLCT001 enabled for replication in HARMDB01

6. Repeat this process for every collection defined in the primary Resource Manager that will be storing objects that you want to make highly available. In our case, this same process is done also for the workstation collection named TSM.NOMIGR.

### 5.3.5 Adjust replication schedule

The replication process runs as a service in every Resource Manager and periodically checks for new objects in the source Resource Manager that need to be replicated to collections in other Resource Managers. One important step that

must be performed is to configure the schedule and frequency with which the replicator will look for candidates and perform the replication.

We need to do this configuration in both the primary and the secondary Resource Managers. The configuration for the replicator process in the primary Resource Manager will be effective for the replication of objects to the standby Resource Manager. The replicator process in the secondary Resource Manager needs to be configured to replicate back objects that were created in the secondary Resource Manager while the primary Resource Manager was down.

To configure the replication process in the primary Resource Manager, use the Content Manager System Administration Client:

1. On the left panel, navigate to the **Configurations** node inside the primary Resource Manager. In our environment, we select **ICMHADB** → **Resource Managers** → **HARMDB01** → **Configurations**.
2. Double-click the **IBMCONFIG** configuration on the right panel and adjust the following values:
  - In the Cycles tab, use the Replicator Hours and Minutes edit boxes to specify the time that the replicator will wait to run a new cycle. This value can vary depending on your requirements. For our scenario, we specify a time of 1 minute between replication cycles. This is the setting we recommend if you want to achieve the best availability at the cost of the greater performance impact on the Resource Managers.
  - In the Replicator Schedule tab, specify which days and for how long the replicator will run. The replicator will only run in the scheduled time frame, with the frequency specified in the Cycles tab. The schedule should also be adjusted according to your requirements. In our scenario, we specify the replicator to run continuously (every day from 00:00 for 24 hours).

**Hint:** Neither the Resource Manager Web application nor the Resource Manager services need to be restarted for these parameters to become effective.

3. After the configuration for the primary Resource Manager is done, perform the same configuration for the secondary Resource Manager. We do this following the same procedure explained before but on the Configuration node inside the secondary Resource Manager. For our environment, we select **ICMHADB** → **Resource Managers** → **HARMDB02** → **Configurations**.

### 5.3.6 Adjust Resource Manager fail-over settings

The Content Manager component responsible for switching operation from a failing primary Resource Manager to the backup Resource Manager is the Library Server monitor. This component runs as a process in the Library Server and periodically checks the status of each Resource Manager to do this.

This component is configured by using the Content Manager System Administration Client:

1. Navigate to the **Configurations** node inside the **Library Server Parameters** branch in the left panel.
2. Double-click **Library Server Configuration** on the right panel.
3. On the Defaults tab, set the Interval to check server availability and the Server timeout. The Library Server monitor checks availability as many seconds as specified in the first parameter, and if one of the Resource Managers does not respond after the time specified for the second parameter, the monitor marks it as unavailable.

The interval can be shortened if you want the automatic failover to occur earlier (thus reducing service down time) at the cost of Resource Manager performance impact. Also, the timeout period should be adjusted in relation to the delay in Resource Manager response during peak activity periods. Default values for both parameters are usually a good choice in most conventional cases. The values set for our sample scenario are shown in Figure 5-30 on page 178.

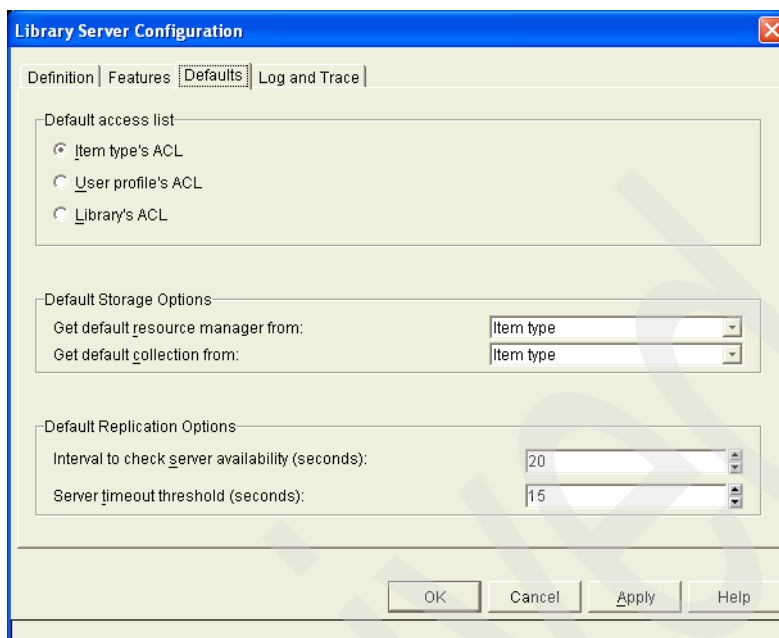


Figure 5-30 Library Server monitor settings

### 5.3.7 Replication test

After the preceding configuration steps are in place, it is a good idea to test that the replication is working. One way to do this is to perform the same steps we did to test if the Resource Manager was working at the end of 5.3.2, “Resource Manager installation and configuration” on page 169. In this case, the steps would be:

1. Define a new item type that stores resource parts on one of the collections enabled for replication with storage into Tivoli Storage Manager.
2. Store a few objects into this item type using one of the Content Manager clients.
3. Verify that you are able to access them by viewing some of them.
4. Query the Tivoli Storage Manager server for files stored by the node name corresponding to the primary Resource Manager and take note of their names.
5. Wait for a few minutes to allow replication to take place.
6. Query the Tivoli Storage Manager server for files stored by the node name corresponding to the standby Resource Manager and compare the names with the ones found in step 4. You should see a file with the same name stored by each node for each object you imported in the system in step 2.
7. Delete the imported objects using one of the Content Manager clients.



8. Let a few minutes pass by to let the migrator and replicator run in the background.
9. Force an inventory expiration in Tivoli Storage Manager.
10. Verify that the files found in step 4 and 6 no longer exist in Tivoli Storage Manager.

## 5.4 The rest of the environment

So far, we described how to set up the Content Manager Library Server and Resource Manager components for high availability. Now, we explain how the rest of the components that compose our test scenario are set up to allow us to test the highly available Content Manager system.

As you can see from Figure 5-1 on page 122, the remaining components consist of the Tivoli Storage Manager server, the eClient mid-tier server, and the rest of the Content Manager clients. Remember that all these components were set up to complete our test scenario, but none of them were set up to be highly available as the Content Manager Library Server and Resource Manager.

### 5.4.1 Tivoli Storage Manager

As explained in “Tivoli Storage Manager server” on page 124, Tivoli Storage Manager is installed on a single separate server and services storage requests for both the primary and the standby Resource Managers.

All the storage in this server is performed on disk pools using the machine’s local disk. There is a separate client node definition for each of the Resource Managers, but they are both grouped together into the same policy domain and management class, so all objects are stored in the same storage pool.

With this in mind, we installed Tivoli Storage Manager server Version 5.2 with the default options. We initialized the server and enrolled the licenses. After that, we used the Web administration interface to create the storage definitions that are summarized in Figure 5-31 on page 180.

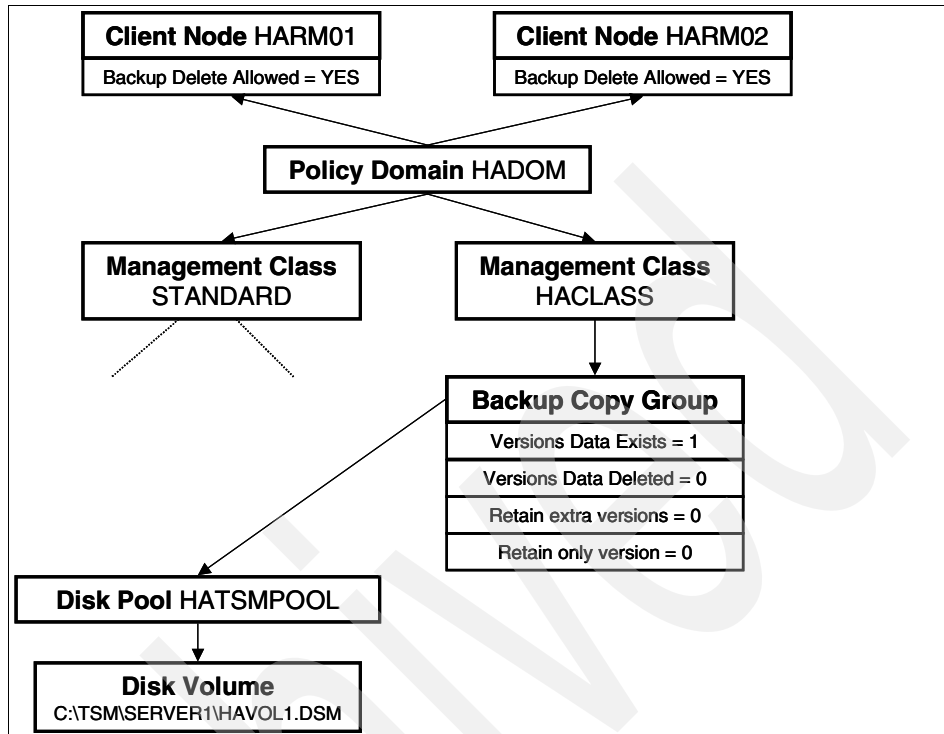


Figure 5-31 Test environment Tivoli Storage Manager definitions for Resource Managers

Using this simple configuration, we know that all objects stored by HARM01 and HARM02 are always stored into C:\TSM\SERVER1\HAVOL1.DSM, which makes it easier for us to check whether the files are stored, replicated, and deleted from Tivoli Storage Manager when we need to test something. Also, from the point of view of the Resource Managers, there is no difference between this simple configuration and a sophisticated, highly available, real-world Tivoli Storage Manager implementation.

## 5.4.2 Content Manager clients

Different Content Manager clients were tested against our highly available Content Manager system installation to see what their reactions were when a system failure occurred. These clients include the Content Manager eClient, Content Manager client for Windows, Content Manager System Administration Client, and a custom API client. All the production clients were installed and configured using default options.

Content Manager eClient server was installed on a separate machine called cm82vm, as shown in Figure 5-1 on page 122. In this machine, we installed the following software products:

- ▶ DB2 UDB ESE Version 8.1 with Fix Pack 3
- ▶ WebSphere Application Server Version 5.0 with Fix Pack 2
- ▶ Information Integrator for Content Version 8.2 with Fix Pack 2
- ▶ Content Manager eClient Version 8.2 with Fix Pack 2

Both the Windows client and the System Administration Client were installed on another machine that was used as an end-user workstation. There, we installed the following software products:

- ▶ DB2 UDB Run-time Client Version 8.1 with Fix Pack 3.
- ▶ Information Integrator for Content Version 8.2 with Fix Pack 2.
- ▶ Content Manager server Version 8.2 with Fix Pack 2 (we chose only the System Administration Client component).
- ▶ Content Manager Windows client Version 8.2.

The last piece in our environment, as shown in Figure 5-1 on page 122, is a small Java program we wrote using Information Integrator for Content Java connector to Content Manager 8.2. We ran this program on the same workstation as the production clients to act as a custom application. This program is detailed in A.2, “Sample custom API program” on page 249.

In both the eClient server and the client workstation, we configured the connection to our Library Server using the following set in Table 5-11.

*Table 5-11 Content Manager client connection information*

| Parameter       | Value              |
|-----------------|--------------------|
| Database name   | ICMHADB            |
| Host name       | euclid.svl.ibm.com |
| Port number     | 60000              |
| Database schema | ICMADMIN           |

## 5.5 Fail-over tests and results

After we performed all the configuration steps explained so far, we had our system up and running ready to service user requests and to stand the failure of the primary Library Server and the primary Resource Manager. In this part, we comment on the different tests performed and the results that were obtained.

Before simulating any kind of failure, we performed a few setup steps common to all tests as summarized in the following.

### Creation of a test item type

We created a simple item type used to store, query, and view documents during the fail-over tests. The only important setting to notice is that the content of the objects stored in this item type is stored in the TSM.NOMIGR collection, as shown in Figure 5-32.

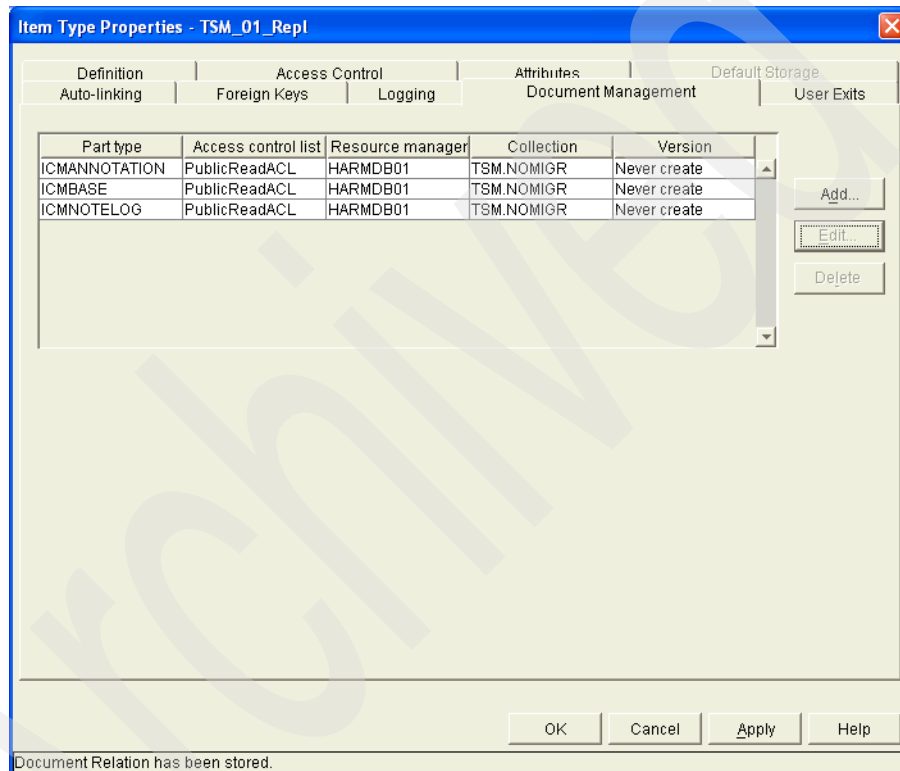


Figure 5-32 Item Type for test runs

#### 5.5.1 Library Server only failover

The first test we performed was to halt the primary server machine (euclid) while the forementioned client components were connected to the system and noted how the overall system behaved.

Our test is as following:

1. Start the client applications.

We started and logged on to the Content Manager System Administration Client, leaving the connection open to the Library Server. We did the same with the Content Manager client for Windows, also leaving an open window displaying a document. We also used a Web browser to log on to the eClient and left the connection open. Finally, we started the Java command-line application, which started importing documents to the system every three seconds.

2. Halt the primary Library Server node.

Logged on as root in a terminal on euclid, we issued the following command to abruptly halt the system:

```
echo "Hello HA" > /dev/kmem
```

This command overwrites critical information on the OS kernel memory, bringing the system down without giving it any chance of cleanup or graceful shutdown. It has the same effect as a power failure, save for the possible hardware problems that can occur on such a case.

3. Library Server fail-over process.

HACMP detects the failure almost immediately and starts the resource group automatically on the secondary node (aristotle). After approximately 35 seconds, the Library Server is up and running again. Of course, any connection that was open to the Library Server at the time of the failure is lost.

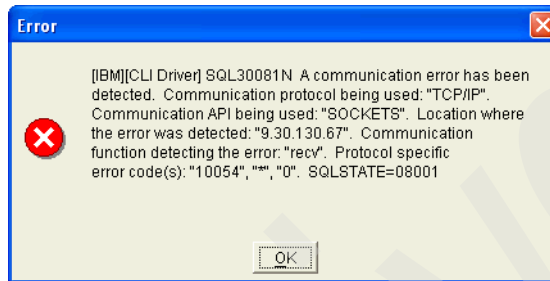
**Note:** The time it takes to have the Library Server available on the standby node will be different in each installation. The main factors that will affect this are the speed of the backup server, the amount of resources allocated for the Library Server database, and the HACMP heart beat settings.

4. Client components behavior.

We observed a different behavior in each of the client components that was using the Library Server at the time of the failure.

In the Content Manager client for Windows, the user starts receiving timeout errors right after the failover occurs. If the user had some work in place (for example, writing a note associated to a document), this might be lost when trying to save. As soon as the Library Server is up again, the user can continue working without the need to log off or reconnect to the server. This Content Manager client component just picks up a new connection to the Library Server and continues working. Of course, if an item was checked out before the failover, it might need to be manually checked in again.

The Content Manager System Administration Client gives a timeout error as soon as the system goes down. After the failover takes place, you still need to close the Content Manager System Administration Client and open it again to go on working with the system. Given this type of client is less frequently used than the other general Content Manager clients, and the fact that usually someone using this interface is a qualified user, we do not think this can cause any inconvenience in normal operation. Figure 5-33 shows the error you might receive if using the Content Manager System Administration Client during a Library Server failover.



*Figure 5-33 CM system administration during Library Server failover*

The eClient also throws an exception as soon as the connection to the Library Server is lost. The message of this exception will be:

```
com.ibm.mm.beans.CMConnectFailedException: DGL0394A: Error in
::DkResultSetCursor.open() [IBM][JDBC Driver] CLI0601E Invalid statement
handle or statement is closed. SQLSTATE=S1000
```

The sessions the users were using are lost. They need to close their browsers and open a new eClient session after the Library Server becomes available again. It is very important that all browsers in the user's workstations are closed to force the eClient to create a new connection for this user.

The sample Java application we wrote was able to recover from the failover after the Library Server became available again. You can see in A.2, "Sample custom API program" on page 249 that this program is designed to retry after a certain amount of time. Also, observe that the datastore object needs to be recycled for the reconnection to take place. Example 5-14 on page 185 is the output that shows the different exceptions received during failover.

*Example 5-14 Output of Java API program during failover*

---

```
C:\CMDaemon>java JImportStuffForBackupTest
Store doc # 0
Store doc # 1
Store doc # 2
Store doc # 3
Store doc # 4
Store doc # 5
Store doc # 6
Store doc # 7
Store doc # 8
Disconnected from datastore. COM.ibm.db2.jdbc.DB2Exception: [IBM][CLI Driver] S
QL30081N A communication error has been detected. Communication protocol bein
g used: "TCP/IP". Communication API being used: "SOCKETS". Location where the e
rror was detected: "9.30.130.67". Communication function detecting the error: "
recv". Protocol specific error code(s): "10054", "*", "0". SQLSTATE=08001
CM Exception recieved. com.ibm.mm.sdk.common.DKDatabaseLogonFailure: DGL0394A:
Error in ::DriverManager.getConnection; [IBM][CLI Driver] SQL30081N A communica
tion error has been detected. Communication protocol being used: "TCP/IP". Comm
unication API being used: "SOCKETS". Location where the error was detected: "".
Communication function detecting the error: "connect". Protocol specific error
code(s): "10060", "*", "*". SQLSTATE=08001 (STATE) : ; [SERVER = icmhadb,
USERID = , SQL RC = -30081, SQL STATE = 08001]
com.ibm.mm.sdk.common.DKDatabaseLogonFailure: DGL0394A: Error in::DriverManager
.getConnection; [IBM][CLI Driver] SQL30081N A communication error has been det
ected. Communication protocol being used: "TCP/IP". Communication API being us
ed: "SOCKETS". Location where the error was detected: "". Communication functio
n detecting the error: "connect". Protocol specific error code(s): "10060", "*"
, "*". SQLSTATE=08001 (STATE) : ; [SERVER = icmhadb, USERID = , SQL RC =
-30081, SQL STATE = 08001]
 at
com.ibm.mm.sdk.server.DKDatastoreICM.connect(DKDatastoreICM.java:2773)
 at
JImportStuffForBackupTest.reconnect(JImportStuffForBackupTest.java:91)
 at JImportStuffForBackupTest.main(JImportStuffForBackupTest.java:68)
Store doc # 9
Store doc # 10
Store doc # 11
Store doc # 12
Store doc # 13
Store doc # 14
Store doc # 15
Store doc # 16
Store doc # 17
Store doc # 18
Store doc # 19
Store doc # 20
Store doc # 21
```

Store doc # 22  
Store doc # 23  
Store doc # 24  
Store doc # 25  
Store doc # 26  
Store doc # 27  
Store doc # 28  
Store doc # 29

---

Finally, the last component that we monitored is the Resource Manager services (replicator, migrator, purger, and stager). Some of these components maintain open connections to the Library Server during normal operation. When a Library Server failover occurs, all the connections to it are lost. All these services run in cycles based on the specified period of time. When it is time for them to run and the connection fails, they write an error message to the appropriate log file and wait for the next scheduled cycle to run again. When the Library Server is available again, the next schedule cycle that runs for each Resource Manager service will reconnect to the Library Server and resume normal operation.

#### 5. Fallback.

After the primary node was operative again, we want to bring the Library Server back to the primary node for the system to be highly available again. As explained in 5.2, “Library Server HACMP” on page 125, the HACMP strategy we choose for the Library Server resource group was *cascading without fallback*. This means that the Library Server resource group has to be manually moved to run on the primary node back again after a failover.

We did this following the recommendations in 5.2.9, “HACMP post-configuration procedures” on page 166 to initiate a resource migration from aristotle to euclid, thus leaving the system ready to stand future failures.

### 5.5.2 Resource Manager failover and fallback

The next test involved the failure and recovery of the primary Resource Manager (harm01vm) while the mentioned Content Manager clients were using the system.

Our test is described as the following:

#### 1. Start the client applications.

Once again, we started and logged on to the Content Manager System Administration Client, leaving the connection open to the Library Server. We did the same with the Content Manager client for Windows, also leaving an open window displaying a document. In addition, we used a Web browser to log on to the eClient and left the connection open. Finally, we started the Java



command-line application, which started importing documents to the system every 3 seconds.

## 2. Verify the Resource Manager replication.

Before simulating the primary Resource Manager failure, we verified that the objects were being replicated to the secondary Resource Manager. To do this, we used the Tivoli Storage Manager Web administration interface to query the contents of the disk volume we defined. The Tivoli Storage Manager command we used for this is:

```
QUERY CONTENTS C:\TSM\SERVER1\HAVOL1.DSM
```

It returned the result shown in Example 5-15.

*Example 5-15 Contents of TSM volume C:\TSM\SERVER1\HAVOL1.DSM*

| Node Name | Type | Filespace Name                 | FSID | Client's Name for File                       |
|-----------|------|--------------------------------|------|----------------------------------------------|
| HARM01    | Bkup | /ICM/HARM-DB01/000-04/HACLA-SS | 2    | \HACCLASS\ L1.A1001001A03K21B60528C078-70.V1 |
| HARM02    | Bkup | /ICM/HARM-DB02/000-02/HACLA-SS | 2    | \HACCLASS\ L1.A1001001A03K21B60528C078-70.V1 |
| HARM01    | Bkup | /ICM/HARM-DB01/000-04/HACLA-SS | 2    | \HACCLASS\ L1.A1001001A03K21B60531D818-51.V1 |
| HARM01    | Bkup | /ICM/HARM-DB01/000-04/HACLA-SS | 2    | \HACCLASS\ L1.A1001001A03K21B60533G898-40.V1 |
| HARM02    | Bkup | /ICM/HARM-DB02/000-02/HACLA-SS | 2    | \HACCLASS\ L1.A1001001A03K21B60531D818-51.V1 |
| HARM02    | Bkup | /ICM/HARM-DB02/000-02/HACLA-SS | 2    | \HACCLASS\ L1.A1001001A03K21B60533G898-40.V1 |

Note that each file exists twice, because it has been stored once by each Resource Manager. This means that replication is taking place normally.

3. Resource Manager failover.

To simulate a failure in the primary Resource Manager, we turned off the power of harm01vm.

After a while, the Library Server monitor marked the primary Resource Manager as unavailable, routing subsequent requests to the secondary Resource Manager, which had replicas of all objects available in the primary Resource Manager. We can see that the failover occurred by using a DB2 command line and querying the Resource Managers table in the Library Server, as shown in Example 5-16. The RMFLAGS column specifies the status of the Resource Manager. A value of 2 means that it is offline, and a value of 0 means that it is online. For further details, refer to “System control tables” in *Content Manager Application Programming Reference*, SC27-1348.

*Example 5-16 Resource Manager fail-over status*

---

```
C:\CMBROOT>db2 connect to icmhadb user icmadmin using password
```

Database Connection Information

```
Database server = DB2/6000 8.1.3
SQL authorization ID = ICMADMIN
Local database alias = ICMHADB
```

```
C:\CMBROOT>db2 select rmname, rmflags from icmadmin.icmstresourcemgr
```

| RMNAME   | RMFLAGS |
|----------|---------|
| -----    | -----   |
| reserved | 0       |
| HARMDB01 | 2       |
| HARMDB02 | 0       |

3 record(s) selected.

```
C:\CMBROOT>db2 connect reset
DB20000I The SQL command completed successfully.
```

---

4. Content Manager clients reaction to the Resource Manager failover.

The reaction of the different Content Manager clients to the Resource Manager failover is even more transparent to the user than the Library Server failover.

For the Content Manager System Administration Client, there is no impact. Note, you cannot do any SMS configuration on the Resource Manager that is down until it is up and running again.

Both the eClient and the Content Manager client for Windows fail to display or store any resource item until failover occurs. After the Library Server monitor

marks the primary Resource Manager as unavailable, both clients resume normal operation, using the secondary Resource Manager instead.

The Java API program also reconnects without major problems. All you have to do is to watch for exceptions, such as the one shown in Example 5-17 (the output of this Java API program), and retry the operation a few times allowing time for the failover to take place.

*Example 5-17 Java API program output during RM fail-over test*

---

```
Store doc # 0
Store doc # 1
Store doc # 2
Store doc # 3
Store doc # 4
CM Exception received. com.ibm.mm.sdk.common.DKException: DGL5161A: Get output
stream for connection to Resource Manager failed. url =
http://harm01vm.svl.ibm.com:80/icrm/ICMResourceManager
Store doc # 5
Store doc # 6
Store doc # 7
Store doc # 8
Store doc # 9
Store doc # 10
Store doc # 11
Store doc # 12
Store doc # 13
Store doc # 14
```

---

Because the primary Resource Manager is down, but operation in the Content Manager system goes on, the imported objects are only stored in the secondary Resource Manager until the primary Resource Manager becomes available again. Example 5-18 is the output of the Tivoli Storage Manager volume contents query after the failover occurred, which reflects this.

*Example 5-18 Tivoli Storage Manager output after failover*

---

| Node Name | Type | Filespace Name                 | FSID | Client's Name for File                       |
|-----------|------|--------------------------------|------|----------------------------------------------|
| -----     | ---- | -----                          | ---- | -----                                        |
| HARM01    | Bkup | /ICM/HARM-DB01/000-04/HACLA-SS | 2    | \HACCLASS\ L1.A1001001A03K21B60528C078-70.V1 |
| HARM02    | Bkup | /ICM/HARM-DB02/000-02/HACLA-SS | 2    | \HACCLASS\ L1.A1001001A03K21B60528C078-70.V1 |
| HARM01    | Bkup | /ICM/HARM-DB01/000-            | 2    | \HACCLASS\ L1.A1001001A03K21B60531D818-51.V1 |

---

|        |      |                                |   |                                              |
|--------|------|--------------------------------|---|----------------------------------------------|
|        |      | 04/HACLA-SS                    |   |                                              |
| HARM01 | Bkup | /ICM/HARM-DB01/000-04/HACLA-SS | 2 | \HACCLASS\ L1.A1001001A03K21B60533G898-40.V1 |
| HARM02 | Bkup | /ICM/HARM-DB02/000-02/HACLA-SS | 2 | \HACCLASS\ L1.A1001001A03K21B60531D818-51.V1 |
| HARM02 | Bkup | /ICM/HARM-DB02/000-02/HACLA-SS | 2 | \HACCLASS\ L1.A1001001A03K21B60533G898-40.V1 |
| HARM01 | Bkup | /ICM/HARM-DB01/000-04/HACLA-SS | 2 | \HACCLASS\ L1.A1001001A03K21B60535J578-34.V1 |
| HARM02 | Bkup | /ICM/HARM-DB02/000-02/HACLA-SS | 2 | \HACCLASS\ L1.A1001001A03K21B60535J578-34.V1 |
| HARM01 | Bkup | /ICM/HARM-DB01/000-04/HACLA-SS | 2 | \HACCLASS\ L1.A1001001A03K21B60538D149-01.V1 |
| HARM02 | Bkup | /ICM/HARM-DB02/000-02/HACLA-SS | 2 | \HACCLASS\ L1.A1001001A03K21B60538D149-01.V1 |
| HARM02 | Bkup | /ICM/HARM-DB02/000-02/HACLA-SS | 2 | \HACCLASS\ L1.A1001001A03K21B60604B582-59.V1 |
| HARM02 | Bkup | /ICM/HARM-DB02/000-02/HACLA-SS | 2 | \HACCLASS\ L1.A1001001A03K21B60606G703-46.V1 |
| HARM02 | Bkup | /ICM/HARM-DB02/000-02/HACLA-SS | 2 | \HACCLASS\ L1.A1001001A03K21B60609C411-19.V1 |
| HARM02 | Bkup | /ICM/HARM-DB02/000-02/HACLA-SS | 2 | \HACCLASS\ L1.A1001001A03K21B60611F549-57.V1 |

---

## 5. Resource Manager fallback.

After the primary Resource Manager is up and running again, the Library Server monitor detects it and marks it as available again in the Library Server by updating the ICMSTRESOURCEMGR table, specifying a value of 0 in RMFLAGS. Then, the replicator service of the secondary Resource Manager replicates back all the objects that were imported while the primary Resource Manager was down. Example 5-19 is the output of the Tivoli Storage Manager volume contents query after the Resource Manager fell back to normal operation.

*Example 5-19 Tivoli Storage Manager output after fallback*

| Node Name | Type | Filespace Name                 | FSID | Client's Name for File                       |
|-----------|------|--------------------------------|------|----------------------------------------------|
| HARM01    | Bkup | /ICM/HARM-DB01/000-04/HACLA-SS | 2    | \HACCLASS\ L1.A1001001A03K21B60528C078-70.V1 |
| HARM02    | Bkup | /ICM/HARM-DB02/000-02/HACLA-SS | 2    | \HACCLASS\ L1.A1001001A03K21B60528C078-70.V1 |
| HARM01    | Bkup | /ICM/HARM-DB01/000-04/HACLA-SS | 2    | \HACCLASS\ L1.A1001001A03K21B60531D818-51.V1 |
| HARM01    | Bkup | /ICM/HARM-DB01/000-04/HACLA-SS | 2    | \HACCLASS\ L1.A1001001A03K21B60533G898-40.V1 |
| HARM02    | Bkup | /ICM/HARM-DB02/000-02/HACLA-SS | 2    | \HACCLASS\ L1.A1001001A03K21B60531D818-51.V1 |
| HARM02    | Bkup | /ICM/HARM-DB02/000-02/HACLA-SS | 2    | \HACCLASS\ L1.A1001001A03K21B60533G898-40.V1 |
| HARM01    | Bkup | /ICM/HARM-DB01/000-04/HACLA-SS | 2    | \HACCLASS\ L1.A1001001A03K21B60535J578-34.V1 |
| HARM02    | Bkup | /ICM/HARM-DB02/000-02/HACLA-SS | 2    | \HACCLASS\ L1.A1001001A03K21B60535J578-34.V1 |
| HARM01    | Bkup | /ICM/HARM-                     | 2    | \HACCLASS\ L1.A1001001A03K21B60538D149-      |

|        |      |                                |                                                |
|--------|------|--------------------------------|------------------------------------------------|
|        |      | DB01/000-04/HACLA-SS           | 01.V1                                          |
| HARM02 | Bkup | /ICM/HARM-DB02/000-02/HACLA-SS | 2 \HACCLASS\ L1.A1001001A03K21B60538D149-01.V1 |
| HARM02 | Bkup | /ICM/HARM-DB02/000-02/HACLA-SS | 2 \HACCLASS\ L1.A1001001A03K21B60604B582-59.V1 |
| HARM02 | Bkup | /ICM/HARM-DB02/000-02/HACLA-SS | 2 \HACCLASS\ L1.A1001001A03K21B60606G703-46.V1 |
| HARM02 | Bkup | /ICM/HARM-DB02/000-02/HACLA-SS | 2 \HACCLASS\ L1.A1001001A03K21B60609C411-19.V1 |
| HARM02 | Bkup | /ICM/HARM-DB02/000-02/HACLA-SS | 2 \HACCLASS\ L1.A1001001A03K21B60611F549-57.V1 |
| HARM01 | Bkup | /ICM/HARM-DB01/000-04/HACLA-SS | 2 \HACCLASS\ L1.A1001001A03K21B60611F549-57.V1 |
| HARM01 | Bkup | /ICM/HARM-DB01/000-04/HACLA-SS | 2 \HACCLASS\ L1.A1001001A03K21B60606G703-46.V1 |
| HARM01 | Bkup | /ICM/HARM-DB01/000-04/HACLA-SS | 2 \HACCLASS\ L1.A1001001A03K21B60604B582-59.V1 |
| HARM01 | Bkup | /ICM/HARM-DB01/000-04/HACLA-SS | 2 \HACCLASS\ L1.A1001001A03K21B60609C411-19.V1 |

At the same time, all the clients continue working without ever noticing they started working with the primary Resource Manager again.

### 5.5.3 Simultaneous Library Server and Resource Manager failover

After these two successful fail-over tests, we performed one last fail-over test consisting of a simultaneous primary Library Server and primary Resource Manager shutdown. The results were equally good.

In short, we simultaneously halted euclid and harm01vm. After approximately 40 seconds, the Library Server was running in aristotle, and all the clients could reconnect. The Resource Manager failover took a few more minutes than the last time, probably because the Library Server had to resume operations first, but it occurred automatically, and the clients could continue working shortly.

Example 5-20 is the output of our Java API program as it stored documents on the Content Manager system when the failure was simulated.

*Example 5-20 Java API program output during Content Manager system failure*

---

```

Store doc # 0
Store doc # 1
Store doc # 2
Store doc # 3
Store doc # 4
Store doc # 5
Store doc # 6
Store doc # 7
Disconnected from datastore. COM.ibm.db2.jdbc.DB2Exception: [IBM][CLI Driver] S
QL30081N A communication error has been detected. Communication protocol bein
g used: "TCP/IP". Communication API being used: "SOCKETS". Location where the er
ror was detected: "9.30.130.67". Communication function detecting the error: "r
ecv". Protocol specific error code(s): "10054", "*", "0". SQLSTATE=08001
CM Exception recieved. com.ibm.mm.sdk.common.DKDatabaseLogonFailure: DGL0394A:
Error in ::DriverManager.getConnection; [IBM][CLI Driver] SQL30081N A communica
tion error has been detected. Communication protocol being used: "TCP/IP". Comm
unication API being used: "SOCKETS". Location where the error was detected: "".
Communication function detecting the error: "connect". Protocol specific erro
r code(s): "10060", "*", "*". SQLSTATE=08001
 (STATE) : ; [SERVER = icmhadb, USERID = , SQL RC = -30081, SQL STATE = 08001]
com.ibm.mm.sdk.common.DKDatabaseLogonFailure: DGL0394A: Error in ::DriverManag
er.getConnection; [IBM][CLI Driver] SQL30081N A communication error has been det
ected. Communication protocol being used: "TCP/IP". Communication API being use
d: "SOCKETS". Location where the error was detected: "". Communication functio
n detecting the error: "connect". Protocol specific error code(s): "10060", "*"
, "*". SQLSTATE=08001
 (STATE) : ; [SERVER = icmhadb, USERID = , SQL RC = -30081, SQL STATE = 08001]
 at com.ibm.mm.sdk.server.DKDatastoreICM.connect(DKDatastoreICM.java:277
3)
 at JImportStuffForBackupTest.reconnect(JImportStuffForBackupTest.java:9
4)
 at JImportStuffForBackupTest.main(JImportStuffForBackupTest.java:71)
CM Exception recieved. com.ibm.mm.sdk.common.DKException: DGL5161A: Get output
stream for connection to Resource Manager failed. url = http://harm01vm.svl.ibm
.com:80/icrm/ICMResourceManager
CM Exception recieved. com.ibm.mm.sdk.common.DKException: DGL5161A: Get output
stream for connection to Resource Manager failed. url = http://harm01vm.svl.ibm
.com:80/icrm/ICMResourceManager

```

.  
. .  
.

CM Exception recieved. com.ibm.mm.sdk.common.DKException: DGL5161A: Get output stream for connection to Resource Manager failed. url = http://harm01vm.svl.ibm.com:80/icrm/ICMResourceManager

Store doc # 8  
Store doc # 9  
Store doc # 10  
Store doc # 11  
Store doc # 12  
Store doc # 13

---





## **Business continuity and disaster recovery strategies**

In this chapter, we introduce the basic concepts and assumptions commonly used to define business continuity and disaster recovery-related concepts. We provide an overview, some background on the need for this focus, and the strategies and options available in Content Manager solutions.

## 6.1 Overview

In this section, we provide an overview for business continuity and disaster recovery. We cover what the disaster recovery plan is, also known as business process contingency plan, and what the objectives and benefits of the plan are. To raise awareness about this topic, we describe some of the most recent disastrous events, the actual cost of the systems down time, the data protection cost in general, and the current trend towards business continuity. Last, we introduce the critical considerations and success factors involved in successfully planning for business continuity and disaster recovery.

### 6.1.1 Business continuity and disaster recovery concept

Disaster recovery is becoming an increasingly important aspect of enterprise computing. Interruption of service or loss of data can have serious financial impact, whether directly or through loss of customer confidence.

A disaster recovery plan (DRP), sometimes referred to as a business continuity plan (BCP), or business process contingency plan (BPCP), describes how an organization is to deal with potential disasters. Just as a disaster is an event that makes the continuation of normal functions impossible, a disaster recovery plan consists of the precautions taken so that the effects of a disaster will be minimized, and the organization will be able to either maintain or quickly resume mission-critical functions. Typically, disaster recovery planning involves an analysis of business processes and continuity needs; it can also include a significant focus on disaster prevention.

As devices, systems, and networks become ever more complex, there are simply more things that can go wrong. As a consequence, recovery plans have also become more complex.

Regulatory compliance is the main reason for implementing a disaster recovery plan. In the last 30 years, the legal requirements for data protection and recovery have been consistently growing to involve today most business sectors. In addition, the demand for business continuity has greatly increased in the past two years. Major catastrophic events have been a “wake up call” for the entire business community and, in particular, for the public sector. The trend of BCP-DRP started in the late 1990s with the preparation for the end of the millennium. Fixing the Y2K bug prompted many organizations to rethink their systems in terms of how best to protect their critical data in case of a system failure on December 31, 1999 at 12:00 midnight. Distributed work environments and transaction-based processing are the types of activities that usually demand the highest level of data protection and BCP-DRP.

## **Objectives and benefits of a BCP-DRP**

Appropriate plans vary from one enterprise to another, depending on variables such as the type of business, the processes involved, and the level of security needed. Disaster recovery planning can be developed within an organization or purchased as a software application or a service. It is not unusual for an enterprise to spend 25% of its information technology budget on disaster recovery.

The objectives and benefits of a BCP-DRP include:

- ▶ Improve storage utilization and data availability.
- ▶ Give users non-stop access and increase storage ROI.
- ▶ Simplify storage management and reduce down time.
- ▶ Increase availability and reduce administration costs.

### ***Improve storage utilization and data availability***

A major benefit of implementing a business continuity plan (BCP) is that the technologies put in place to meet BCP and DRP requirements help streamline operations and rationalize the use, storage, and disposition of business data depending on how critical that data is to the organization.

### ***Give users non-stop access and increase storage ROI***

Another benefit is to provide users with uninterrupted access to data and content through the implementation of high availability technologies, improving the service provided by the solution. In addition, a DRP often increases the Return on Investment (ROI) of a storage solution by decreasing system down time and avoiding the cost of data loss.

### ***Simplify storage management and reduce down time***

Most of the time, the solution implemented allows you to simplify storage management procedures and highly reduce, in some cases eliminate, end-user down time. This is another contribution to increased ROI and (end-user community) customer satisfaction.

It is possible to simplify storage management by using a global strategy (holistic approach) to address common disaster recovery requirements across the organization. Following this principle, all applications and systems (departmental and organization-wide) should comply with the same disaster recovery requirements and guidelines based on the different type of data processed by those systems.

### ***Increase availability and reduce administration costs***

An additional objective and benefit of implementing a DRP is to increase system availability by leveraging high availability technology and strategies that contribute to improving the continuity of operations.

Administration costs can be greatly reduced by addressing the disaster recovery issue at an organizational level and by standardizing hardware and software components. This will allow centrally administration of the entire organization's resources while implementing a distributed solution when applicable.

Asset management software provides this capability, resulting in a reduced number of system administrators across the organization.

## **6.1.2 Major disastrous events**

Business continuity plans must account for power disruption and other natural or man-made disastrous events and architect alternate sites at sufficient distance separation. Table 6-1 lists some of the recent major natural and man-made disasters.

*Table 6-1 Examples of the recent major natural and man-made disasters*

| <b>Location: Event</b>   | <b>Description</b>                                                                                                                                           | <b>Impact</b>                                                                                                                     |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| U.S.: Northeast blackout | On August 14, 2003 at 4:00 p.m., on a work day, a massive power outage affected major cities in the Eastern power grid, hundreds of miles distant.           | Collapse of mass transportation system, major traffic congestion, and loss of business activities for several days.               |
| Italy: Total blackout    | On September 28, 2003, the entire country suffered the worst power outage since World War II. <sup>a</sup>                                                   | Most of Italy's 58 million people and businesses were affected.                                                                   |
| U.S.: Hurricane Isabel   | On September 18, 2003, the most powerful U.S. hurricane in over a decade hit the mid-Atlantic East coast. Government and businesses were shuttered for days. | Close to 6 million people were without power. Some places were without power for up to 5 days. Losses in the billions of dollars. |

| Location: Event    | Description                                                                               | Impact                                                                                                                                                    |
|--------------------|-------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| U.S.: New York, NY | On September 11, 2001, a terrorist attack stroke the World Trade Center in Manhattan, NY. | Extensive loss of personnel and complete destruction of facilities, disruption of governmental services that ensure the general public health and safety. |

a. Believed to be a British tree uprooted by strong German wind hitting a Swiss/Italian transmission line; that could have caused overcapacity on French lines to Italy, resulting in a domino-like collapse.

The cause of the blackout illustrates the precarious state of national power grids and international interdependence.

The lasting effect of 9/11 is a change in awareness of the potential magnitude of a disaster:

- ▶ Now, extensive loss of personnel and complete destruction of facilities has proven a reality.
- ▶ People have become keenly aware that lapses in mission-critical operations have dire cascading effects throughout the economy.
- ▶ This is especially true when those operations are governmental services that ensure the general public health and safety.

### 6.1.3 Data protection cost

The additional cost associated with data protection varies from one organization to another and from one type of solution to another. The key questions to address when planning for business continuity are:

- ▶ How much data can the organization afford to lose?
- ▶ What is the organization's recovery point objective (RPO)?
- ▶ How long can the organization afford to have the system offline?
- ▶ What is the organization's recovery time objective (RTO)?

#### Factors contributing to higher costs

There are many factors contributing to higher costs. They include:

- ▶ More complex IT operating environment as a result of exponential growth of storage capacity and diversification of operating systems (strong growth of Windows NT® in the last two decades and the emergence of Linux in the late 1990s).

- ▶ The new era of e-business requires solutions operating at 24 hours a day, 7 days a week, 365 days a year (7x24x365), and on a more global basis of information exchange between dispersed sites.
- ▶ Digital data continues to expand dramatically, and more data is critical to the business.
- ▶ Increased complexity with new data protection and regulatory requirements.
- ▶ Confusing set of point product options available.

#### 6.1.4 Trends in business continuity planning

Large organizations are showing more interest in BCP-DR. Skill transfer and training is still underestimated. Simulations, or rehearsals, are needed for gap identification. Spending, however, remains moderate.

##### **New facts about business continuity**

Business are setting the service-level bar higher. Some of the new facts about business continuity include:

- ▶ Traditional 72-hour recovery periods for business-critical processes are not good enough anymore.
- ▶ A general new 4-24 hour recovery time and point objectives.
- ▶ A need for a larger goal of ensuring resumption and recovery of end-to-end enterprise business processes.
- ▶ Active/passive configuration between two sites for 30-60 minute recovery.
- ▶ The 24x7 continuous availability being designed into most critical applications.

##### **Geographic diversity is imperative**

One of the new trends is the goal of geographically diversifying the primary and the backup sites and different infrastructure components. The 9/11 experience expanded the definition of “significant” distance between sites. The U.S. Securities and Exchange Commission (SEC) is now mandating same day recovery for firms that play critical roles in financial markets. Other compliance regulations favor a network approach.

For example:

- ▶ Campus/local  $\leq 10$  km  
Campus size or local, distance is less than or equal to 10 kilometers (km).
- ▶ MAN = 80-100 km  
Medium area network size, distance is between 80 to 100 kilometers.

- ▶ WAN  $\geq$  1000s km  
Wide Area Network, distance is greater than or equal to 1,000 kilometers.

### 6.1.5 Critical considerations and success factors

Critical considerations for various business models and processing include:

- ▶ Information-based business model  
Highly digital versus paper-based business. 24x7 access to information is critical. Security of information is critical.
- ▶ Transaction-based (versus batch-based) processing  
Operations stop when IT is down.
- ▶ Distributed work environment  
Mobile workforce. Remote offices need to tie to headquarters.
- ▶ People-based business  
Implying, productivity of employees closely tied to access to technology and information.

Success factors include:

- ▶ Holistic approach to solution design.
- ▶ Emphasis on integrating “islands of automation.”

Solution design integrating business continuity principles include:

- ▶ Categorize applications to determine requirements.
- ▶ Determine which applications and data are critical.
- ▶ Know your cost of down time for these applications.
- ▶ Develop a solution that considers a combination of need and cost.
- ▶ Implementation period can be long.
- ▶ Periodic testing is mandatory.
- ▶ Compliance to data center disaster recovery strategy within overall corporate business continuity objectives.

## 6.2 Business continuity strategies and options

Customer options can be categorized in two groups: insourced and outsourced.

## 6.2.1 Insourced

A customer invests in building an internal backup infrastructure that the customer owns and manages, as well as provisions all the necessary staffing and service requirements, including managing third-party providers, such as network service providers and “redundant” power suppliers.

## 6.2.2 Outsourced

Customers leverage a third-party provider that both delivers business continuity/disaster recovery services and manages the delivery of services from the provider's own infrastructure (for example, data centers and mobile units) for two types of customer options:

- ▶ **Dedicated**

In a dedicated environment, customers have access to a set of systems/storage reserved exclusively for a single customer.

- ▶ **Shared**

In a shared environment, customers have access to a set of systems/storage that more than one customer can leverage.

## 6.2.3 Business continuity needs assessment

You need put in place a needs assessment together with an action plan. Prior to putting in an action plan, you must first assess the business continuity pertaining to your environment:

- ▶ **What is the need for business continuity in your business?**
  - Does your business need to access data 24x7?
  - Is your data vulnerable to security attacks?
  - Is your IT processing transaction-based (versus batch)?
  - Is your business made up of a large mobile workforce?
  - Are there many remote locations?
  - Is the business model very centralized?
- ▶ **What is the cost-per-hour of down time in your business?**
  - By functional area (for example, by HR, accounting, and call center)
  - By IT/telecom area (for example, data center, network, desktop systems, applications, and telecommunications)
- ▶ **Consider the pros and cons of an insourced versus outsourced model.**



## 6.3 Disaster recovery best practices

Planning for disaster recovery of enterprise content management solutions is different from database, or most business-centric applications, because it involves different software and hardware components and technologies. Most content management solutions include data, image, Web content, records management, and business integration services. This implies that all the underlying components of the solution (for example, database, application server, and services) will need to be recovered in the case of a disaster and need to be synchronized to maintain data integrity. Applying data protection and recovery, specific to each component, might not always be the most cost-effective or workable solution in this case.

In general terms, the choice of the appropriate disaster recovery technology to be used is, in most cases, driven by the customer's cost-benefit perception. That is, the cost of protecting, and eventually recovering, critical data versus the added benefits provided by disaster recovery technology previously discussed in "Objectives and benefits of a BCP-DRP" on page 197.

Depending on the type of data requiring protection, different disaster recovery technology with various cost implications are available in the market today. Data can be categorized in five types, or classes, following its recovery point objective (RPO) and its recovery time objective (RTO):

- ▶ Class 1 and 2 data: The RPO and RTO are measured in days for this type of data. A local tape backup solution will provide the appropriate cost-benefit relationship for this type of data.
- ▶ Class 3 data: The RPO is measured in hours, and RTO is measured in days for this type of data. A remote tape will be the less costly solution, and disk replication taking snapshots of the data will be the high-end cost for a recovery solution meeting the requirements for this type of data.
- ▶ Class 4 data: The RPO is measured in seconds, and the RTO is measured in hours for this type of data. An asynchronous disk replication recovery solution will meet the RPO requirements, and a virtual tape solution would meet the RTO requirements for this type of data.
- ▶ Class 5 data: The RPO is measured in seconds, and the RTO is also measured in minutes for this type of data. A synchronous disk replication recovery solution would meet the RPO and the RTO requirements for this type of data.

Figure 6-1 on page 204 represents the relationship between the different data classes, the RPO and RTO, and the total cost of ownership (TCO) of the solution required following each case. For example, when the requirements for RTO is "continuous" service, the technology to use is synchronous disk replication; tape

technology is not a good choice. At the same time, the TCO of the solution is the highest.

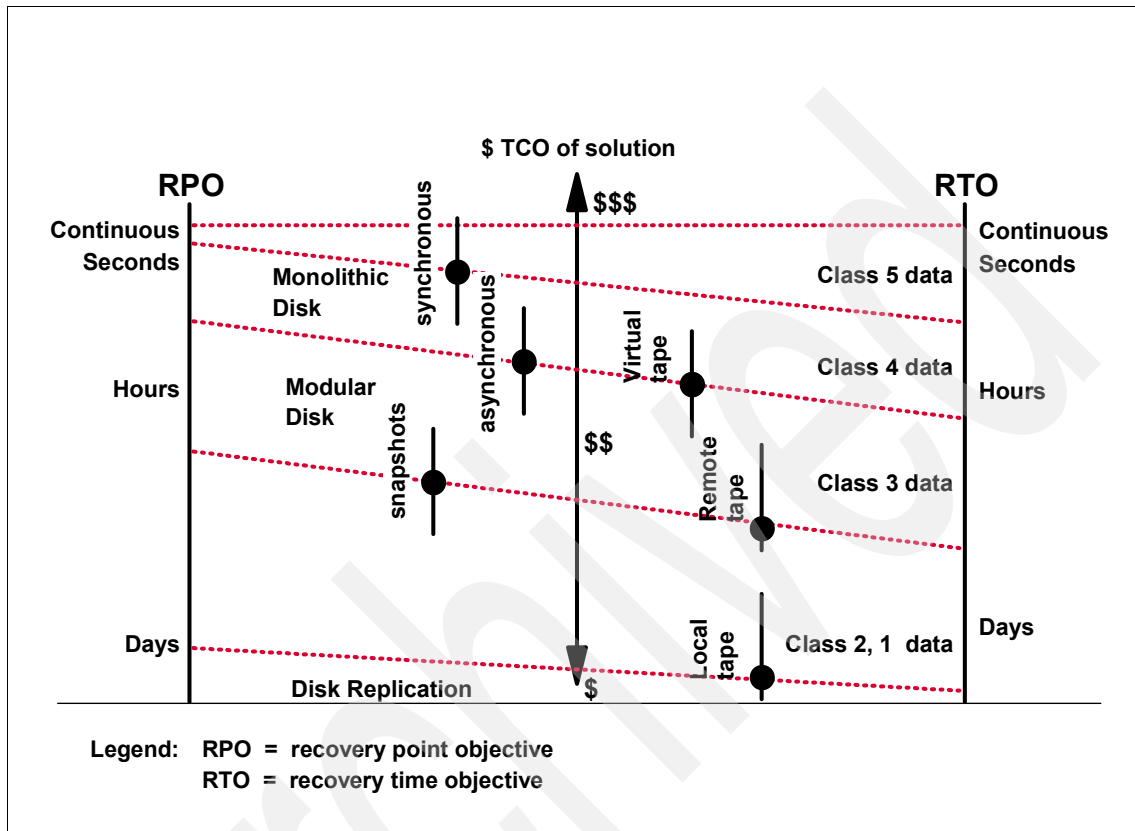


Figure 6-1 RPO and RTO for different data classes, solutions, and total cost of ownership (TCO)

Figure 6-2 on page 205 describes the different levels, or tiers, of disaster recovery based on their cost/recovery time relationship and the technology used to meet the requirements in each case.

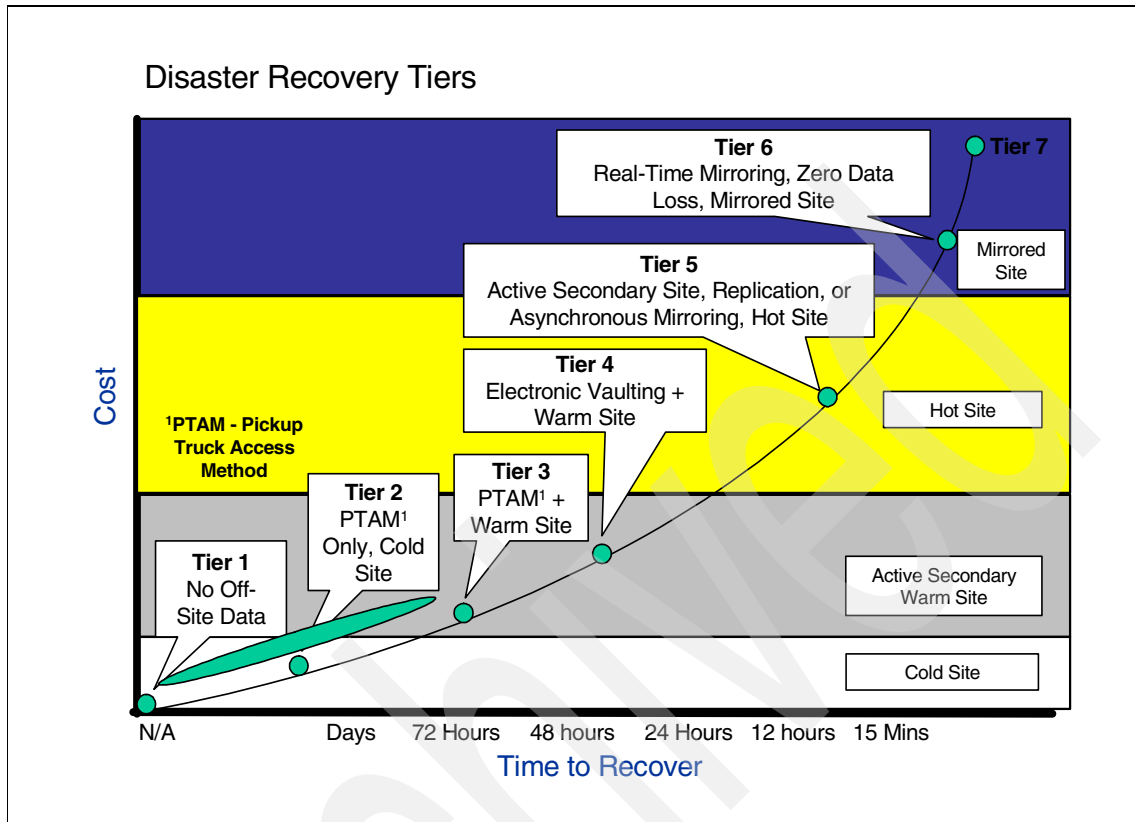


Figure 6-2 The seven tiers of DR solutions in function of the cost/time to recover relationship

### 6.3.1 Disaster recovery: Best practices with Content Manager

Content Manager Version 8 provides some options and strategies to help meet the stringent DRP requirements providing metadata and content protection, enhanced availability, and mechanisms for recovery and synchronization.

#### Business considerations

The cost factor that most businesses emphasize is the cost of implementing a disaster recovery solution versus the cost of losing the data and the system down time. When considering cost, it is important to include the *labor*, *time*, and *business cost* of recreating the data, the content, and the systems if the entire solution were to be destroyed. In addition to this partial list of tangible costs, there are the *intangibles*. If the customer's perception is that the data and content cannot be rebuilt as it was originally, there might be a loss of confidence, and even business.

## Technical considerations

There are specific components and functions Content Manager provides to facilitate the recovery of lost data, content, and system functionality. If designed and configured properly, Content Manager solution protects the business logic of the system and allows organizations to remain operational soon after a disaster.

A disaster recovery strategy implies the implementation of resources and mechanisms to allow for the recovery of all system components in case of the loss of the ability to run the system due to unplanned catastrophic events. That means using a combination of additional software, hardware, hosting site, and human resources, which should be immediately available when the main system fails to resume its operations.

### ***Application availability and system backup***

Designing a Content Manager solution that includes provisions for *enhanced availability*, or *high availability*, and a *comprehensive system backup strategy* is an important strategy for implementing business continuity and disaster recovery.

Enhanced availability of a solution, supported through the redundancy of functionality and components built in the design, constitutes a step towards recovering the system within minutes of a system failure. A replica of the Content Manager main Library Server can be configured and maintained operational at a different site. In case of an application failure, logging out from the main Library Server and logging in to the remote Library Server is required to fulfill manual failover and ensure business continuity. This can be accomplished within minutes. Figure 6-3 on page 207 and Figure 6-4 on page 208 show a simple Content Manager Version 8.2 configuration that meets the requirements referred to as disaster recovery tier 5 (see Figure 6-2 on page 205) with an active secondary site that includes the following:

- ▶ Asynchronous “replication.”
- ▶ Content Manager and DB2 replication (transaction unaware).
- ▶ Execution Content Manager Synchronization utility on demand or at scheduled hour.
- ▶ When a disaster is declared, the entire environment is switched.
- ▶ The secondary site can be configured with, or without, HA.

Site A is composed of four layers, or tiers (presentation, authentication/security, application, and data repository). Site A has a fail-over site, Site B. Site B has the same layers of functionality and configuration as described in Site A and is made available in case of a failure on Site A.

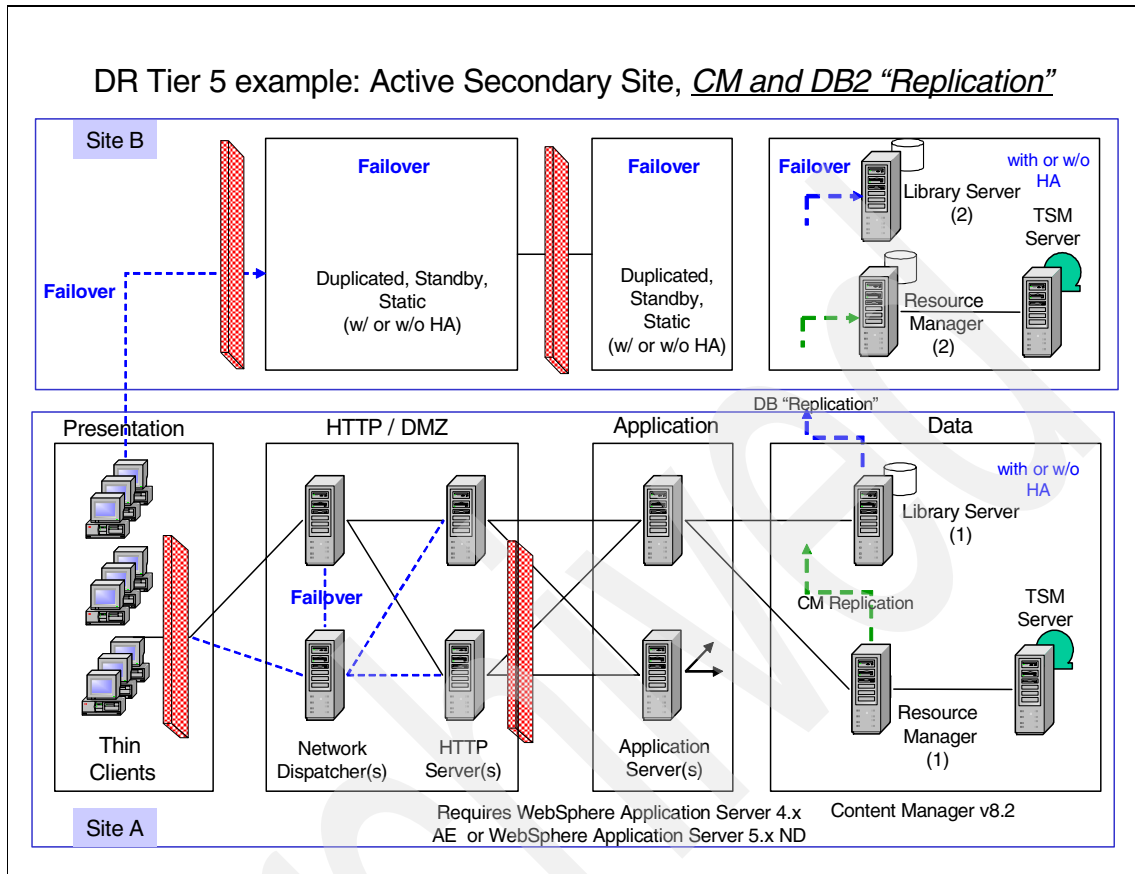


Figure 6-3 Example of DB2 Content Manager DR with on-site and geographical failover

This configuration can be enhanced by using operating system HA technology such as HACMP on AIX, Sun Solaris cluster, Windows NT cluster, z/OS clustering with sysplex technology, or similar. In this case, the solution would leverage generic HA and Content Manager content replication.

System backups are essential in recovering data and content. In a Content Manager solution, it enacts through a dedicated Tivoli Storage Manager server backing up the system configuration to an LTO tape backup system. The Tivoli Storage Manager server can be scheduled to perform full systems backups and incremental backups, or both.

A third strategy consists of implementing a Storage Area Network (SAN). For example, by leveraging the EMC Timefinder and BCV capabilities, full backup of the operating system, file structure, database, and images can be performed weekly with little to no interruption to the continuous operation of the production

system. Also, an incremental backup of modified data and images since the last full backup can be performed daily without affecting the continuous operation of the production system.

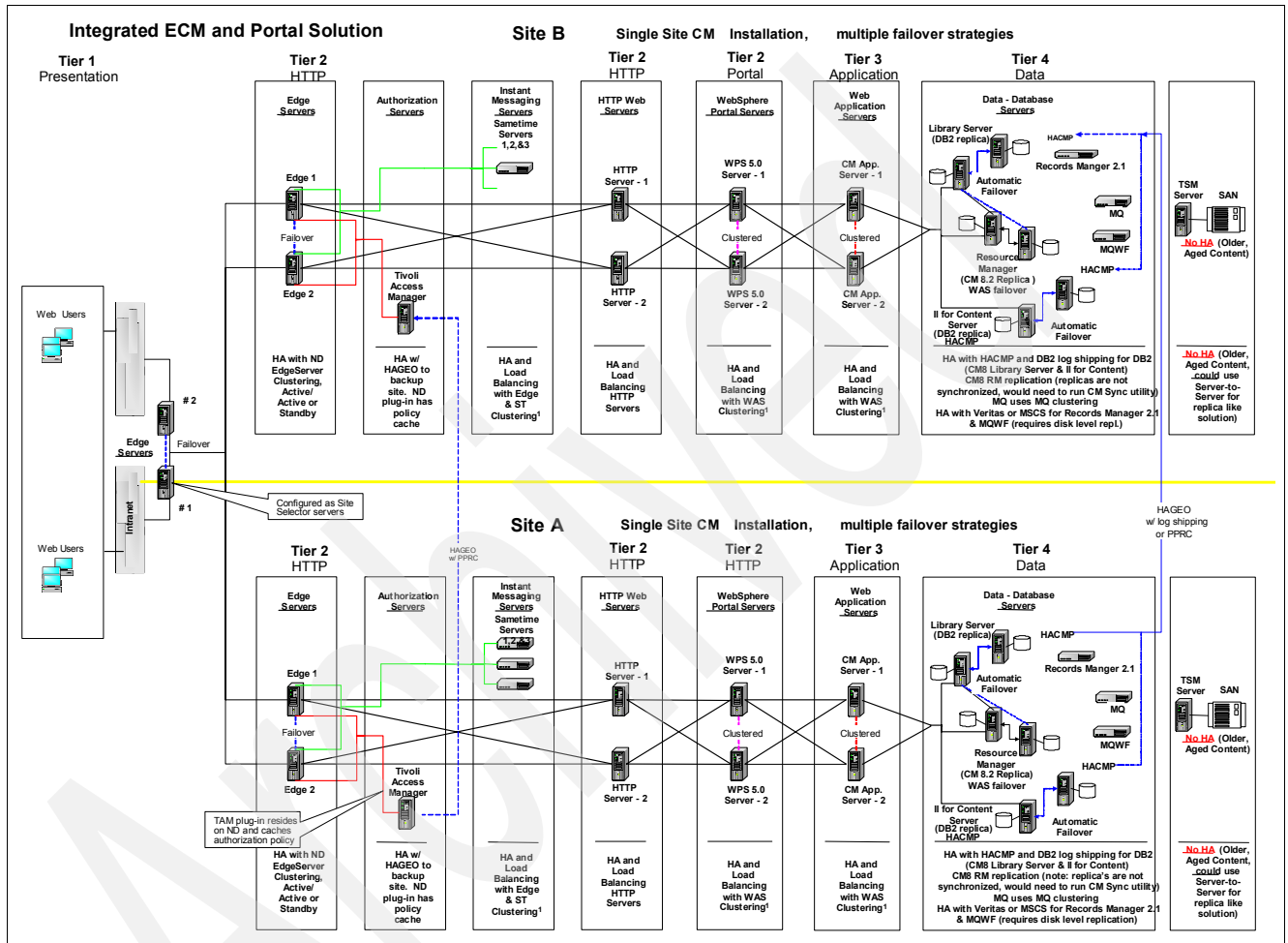


Figure 6-4 Detailed diagram with local and geographical failover of a Content Manager solution

### 6.3.2 Content Manager recovery

There are several Content Manager utilities that you can use to provide recovery:

- ▶ Asynchronous recovery utility
- ▶ Resource Manager/Library Server validation utility
- ▶ Resource Manager volume validation utility

## Asynchronous recovery utility overview

Content Manager includes an automatic scheduled process called the asynchronous recovery utility. We call it an automatic scheduled process, because the migrator process will automatically run this utility, and the migrator process is scheduled. Its purpose is to periodically restore data consistency between a Library Server and its Resource Managers. This process is necessary for the following reasons:

- ▶ To provide a rollback function for failed transactions
- ▶ To complete scheduled deletion of items that are designated for deletion
- ▶ To delete tracking table records (for both the Library Server and the Resource Manager) for transactions that are determined to have completed successfully

The Library Server and Resource Manager can become inconsistent in the event that the Resource Manager crashes or communications between the Information Integrator for Content API calls and Resource Manager fails. The inconsistent state can be reconciled with the asynchronous transaction reconciliation utility.

**Note:** Before performing any work, the migrator process will first run the asynchronous recovery utilities.

Another important result of running this utility is to clean up known successful transactions. As each create/update resource item transaction completes, a record is placed into the Library Server database. These records and their database table become larger over time. The table is cleaned up by the transaction reconciliation utility. It is important to run the utility on all of the Content Manager Resource Managers.

Also, deletion of Resource Manager resources is an asynchronous activity within Content Manager. When a user uses an application to delete an item, it is deleted, internally, from the Library Server. The asynchronous recovery deletion reconciliation utility is used to mark or physically delete the resource on the Resource Manager. The resource deletion is a multiple step process. On Windows, AIX, and Solaris platforms, the Resource Manager migrator, running in the background, is responsible for taking all of the resources marked for deletion and physically deleting them. Resource deletion consists of three steps:

1. A Content Manager application deletes an item from the Library Server.
2. The asynchronous recovery deletion reconciliation utility marks the resource for deletion on the Resource Manager.
3. The Resource Manager migrator physically deletes the resource.

Although these processes are scheduled and automatic processes, you might want to run the programs themselves, for example, as part of a database backup procedure. To do so, you need to execute two commands to run two separate utility programs:

- ▶ The deletion reconciliation utility (ICMRMDEL)
- ▶ The transaction reconciliation utility (ICMRMTX)

**Tip:** In a production environment, synchronize the servers prior to any system backup. This not only ensures your databases are in a consistent state, but also removes any database entries that represent deleted documents.

### ***Configuring the asynchronous recovery utility***

The asynchronous recovery stand-alone utilities use the `icmrepenv.sh` (for AIX and Sun Solaris) or the `icmrepenv.bat` (for Windows) file for specifying the WebSphere directories when installing the Resource Manager. These files, found in the `%ICMROOT%/config` directory, are also used in specifying the `DB2Instance`, location of the DB2 JAR files, and Oracle JAR files. These files also enable the use of WebSphere Version 5. Using these files is a change from the Content Manager Version 8.1 asynchronous recovery utilities, where the `rmpath` and `DB2Instance` are optional input parameters.

### ***Asynchronous utility logging***

By default, the asynchronous utilities log to the console. You can modify the level of information logged and the location of the output in the `icmrm_asyncr_logging.xml` file. This XML file can be updated to output to FILE if desired. Make sure that the user ID that you use to run the utility has read permission to the XML file and write permission to whatever log file that you configure for use.

The `icmrm_asyncr_logging.xml` file is installed with the Resource Manager code in the WebSphere Application Server `installedApps` path.

On AIX, the default path to the file is:

```
/usr/WebSphere/AppServer/installedApps/icmrm.ear/icmrm.war
/icmrm_asyncr_logging.xml
```

On Solaris, the default path is:

```
/opt/WebSphere/AppServer/installedApps/icmrm.ear/icmrm.war
/icmrm_asyncr_logging.xml
```

On Windows, the default path is:

```
x :\\WebSphere \\AppServer \\installedApps\\icmrm.ear\\icmrm.war
\\icmrm_asyncr_logging.xml
```



### ***Running the asynchronous recovery utilities on Windows***

To run the two asynchronous recovery utilities:

1. Open a command prompt window.
2. Enter `icmrmdel.bat` to run the deletion reconciliation utility.
3. Enter `icmrmtx.bat` to run the transaction reconciliation utility.

### ***Running the asynchronous recovery utilities on AIX***

To run the two asynchronous recovery utilities:

1. From a command prompt, enter `cd /usr/lpp/cmb/bin`.
2. Enter `icmrmdel.sh` to run the deletion reconciliation utility.
3. Enter `icmrmtx.sh` to run the transaction reconciliation utility.

### ***Running the asynchronous recovery utilities on Solaris***

To run the two asynchronous recovery utilities:

1. From a command prompt, enter `cd /opt/IBMicm/bin`.
2. Enter `icmrmdel.sh` to run the deletion reconciliation utility.
3. Enter `icmrmtx.sh` to run the transaction reconciliation utility.

**Tip:** After running the asynchronous recovery utilities, run the RUNSTATS function on your database to ensure that they are operating efficiently.

## **Overview of validation utilities**

The purpose of the validation utilities is to analyze discrepancies between three components: the Library Server, the Resource Manager, and the storage system or systems used by the Resource Manager through its defined device managers. Any of these components can fail and require a restoration through a backup that might be out of synchronization with the other two components.

Because there is no direct link between the Library Server and the storage system (an example of a storage system is VideoCharger or Tivoli Storage Manager), differences must be reported between the Library Server and the Resource Manager, and the Resource Manager and the storage system, using the following utilities:

- ▶ The *Resource Manager/Library Server validation utility* (`icmrmlsval.sh` or `icmrmlsval.bat`) generates reports that describe discrepancies between the Library Server and the Resource Manager.
- ▶ The *Resource Manager volume validation utility* (`icrmrmvolval.sh` or `icrmrmvolval.bat`) generates reports on discrepancies between the Resource Manager and the storage system.

The reports are in XML. You can use commonly available XML tools or browsers to view or manipulate the utility output files. Content Manager installs the XML DTD required by the validation utility output files.

You can modify the two utility files with information specific to your Content Manager system. The validation utilities are located in the bin directory in the Resource Manager installation directory.

The validation utility creates and drops a temporary DB2 table. The environment script requires the resource database name, user ID, password, schema, Web application path, and DB2 instance. To set the environment for both validation utilities, type:

```
setenvproc.bat OR setenvproc.sh
```

### ***Logging***

By default, the validation utilities log to a file named icrmr.validator.log in the WebSphere logs directory. You can modify the level of information logged and the location of the output in the icrmr\_validator\_logging.xml file. Be sure that the user ID that you use to run the utility has read permission to the XML file and write permission to whatever log file that you configure for use.

The icrmr\_validator\_logging.xml file is installed with the Resource Manager code in the WebSphere Application Server installedApps path.

On AIX, the default path to the file is:

```
/usr/WebSphere/AppServer/installedApps/icrmr.ear/icrmr.war
/icrmr_validator_logging.xml
```

On Solaris, the default path is:

```
/opt/WebSphere/AppServer/installedApps/icrmr.ear/icrmr.war
/icrmr_validator_logging.xml
```

On Windows, the default path is:

```
x:\WebSphere\AppServer\installedApps\icrmr.ear\icrmr.war
\icrmr_validator_logging.xml
```

### **Working with the RM/LS validation utility**

The Resource Manager/Library Server validation utility queries the Library Server for all of the objects created or updated in a specified time period. It then searches the Resource Manager database and detects any discrepancies. The utility runs on the Resource Manager server and requires connectivity to the Library Server database.

To start the utility, navigate to the Resource Manager bin directory and type:

`icmrmlsval.sh` or `icmrmlsval.bat`

The utility requires input parameters that are described in Table 6-2. Both dashes (-) and forward slashes (/) are handled as the parameter separator. The parameter tags are supported in both lowercase and uppercase.

*Table 6-2 Resource Manager/Library Server validation utility parameters*

| Parameter              | Description                                                                                                                                                                                                                                                                                                                            |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -B YYYY-MM-DD-HH.MM.SS | The beginning time and date of the objects to examine. Use this parameter with the -E parameter to restrict the number of objects that the utility must examine. This parameter is optional. If it is not present, all of the objects prior to the -E date are returned, or all of the objects are returned if -E is also not defined. |
| -E YYYY-MM-DD-HH.MM.SS | The ending time and date of the objects to synchronize. Use this parameter with the -B parameter to restrict the number of objects that the utility must examine. This parameter is optional. If it is not present, all of the objects after the -B date are returned, or all of the objects are returned if -B is also not defined.   |
| -F output-path         | The absolute path to be used for the output files. The utility creates the UTF-8 XML files in this directory. This parameter is required.                                                                                                                                                                                              |
| -H                     | This parameter displays help information about how to invoke the utility. All of the other parameters are ignored, and no processing occurs.                                                                                                                                                                                           |

The utility creates a temporary table, `RMLSITEMS`, used to accumulate object statistics for the validation. At the end of the validation, this table is normally dropped. If the utility determines that the table is present, it presumes another version of the utility is operating and exits. If the table is left behind due to an aborted run, you need to drop this table. Connect to the Resource Manager database and drop the table with the following command:

```
db2 drop table RMLSITEMS
```

The following line shows an example of how to invoke the Resource Manager/Library Server utility on an AIX server:

```
./icmrmlsval.sh -F /reportsdirectory -B 2002-08-30-00.00.00 -E
2002-09-01-00.00.00
```

## ***Understanding the Resource Manager/Library Server reports***

The base file names of the reports are “icmrmlsval YYMMDDHHMMSS \_”+Report Type string +”.xml”. The Report Type string identifies the type of discrepancies a report contains. The description of the different report types are detailed in this section. The time stamp allows the administrator to run the utility multiple times without overwriting the output files. Examples of default names with the default report type are:

- ▶ cmrmlsval20020531123456\_ORPHAN.xml
- ▶ cmrmlsval20020531123456\_NOTINRM.xml
- ▶ cmrmlsval20020531123456\_SIZEMISMATCH.xml
- ▶ cmrmlsval20020531123456\_COLLECTIONMISMATCH.xml
- ▶ icmrmlsval20020531123456\_DATEMISMATCH.xml

There are several types of Resource Manager/Library Server reports:

### **Orphan**

Entries are added to the ORPHAN report if an object is on the Resource Manager, but the Library Server does not have a reference to the object. The report contains information about the object from the Resource Manager database.

### **Not in RM**

Entries are added to the NOTINRM report if the Library Server has a reference to an object, but the object is not on the Resource Manager. The report contains information about the object from the Library Server database.

### **Size mismatch**

Entries are added to the SIZEMISMATCH report if the size of an object on the Library Server does not match the size of an object on the Resource Manager. The report contains information about the object from the Resource Manager and Library Server databases.

### **Collection mismatch**

Entries are added to the COLLECTION report if the collection of an object on the Library Server does not match the collection of an object on the Resource Manager. The report contains information about the object from the Resource Manager and Library Server databases.

### **Date mismatch**

Entries are added to the DATEMISMATCH report if the object update date on the Library Server does not match the object update date on the Resource Manager. Under normal circumstances, if there is any synchronization problem between the Library Server and the Resource Manager, the object update date does not match. In order to reduce redundant entries in the different reports, entries are not added to the DATEMISMATCH report if they have been added to the collection mismatch or size mismatch reports. The report contains information about the object from the Resource Manager and Library Server databases.

### **Working with Resource Manager volume validation utility**

The Resource Manager volume validation utility checks each object in its database that was added or changed in a specified date range. It queries the device manager for the attributes of that object and generates reports for each object whose information in the database is different than reported by the device manager. You might want to use the utility if you have to restore data on a volume after a volume crash. The utility helps you to verify that the data is restored correctly. The Resource Manager must be running when you use the utility.

**Tip:** Use the Resource Manager volume validation utility during times of low traffic on the Resource Manager.

The validation utility does not search the storage system for orphaned objects (objects not referenced by the Resource Manager). Because there are a wide variety of storage systems that are often used for storing files other than those managed by Content Manager, the scanning for orphaned files can be extremely time consuming and can produce a large quantity of false positives.

The Resource Manager volume validation utility runs on the Resource Manager server and only requires access to its own database and the device managers responsible for the volumes that are being checked.

### ***Starting the Resource Manager volume validation utility***

The Resource Manager volume validation utility is `icrmrmvalval.sh` or `icrmrmvalval.bat`. To start the utility, navigate to the bin directory in the Resource Manager home directory.

The Resource Manager volume validation program uses specific input parameters (see Table 6-3 on page 216). Both dashes (-) and forward slashes (/) are handled as the parameter separator. The parameter tags are supported in both lowercase and uppercase.

Table 6-3 Resource Manager volume validation utility parameters

| Parameter              | Description                                                                                                                                                                                                                                                                                                                            |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -B YYYY-MM-DD-HH.MM.SS | The beginning time and date of the objects to examine. Use this parameter with the -E parameter to restrict the number of objects that the utility must examine. This parameter is optional. If it is not present, all of the objects prior to the -E date are returned, or all of the objects are returned if -E is also not defined. |
| -E YYYY-MM-DD-HH.MM.SS | The ending time and date of the objects to synchronize. Use this parameter with the -B parameter to restrict the number of objects that the utility must examine. This parameter is optional. If it is not present, all of the objects after the -B date are returned, or all of the objects are returned if -B is also not defined.   |
| -F output-path         | The absolute path to be used for the output files. The utility creates the UTF-8 XML files in this directory. This parameter is required. If the files currently exist, they are overwritten.                                                                                                                                          |
| -H                     | This parameter causes the program to display help information about how to invoke the utility. All of the other parameters are ignored and no processing occurs.                                                                                                                                                                       |
| -V volume-name         | The logical volume name on which you want to perform the validation. Use this parameter to limit the number of storage systems to one volume. This parameter is optional. If not used, all storage systems are searched.                                                                                                               |

### ***Understanding the validation discrepancy reports***

The base file names of the discrepancy reports are “icmrmvolvalYYMMDDHHMMSS\_” + Report Type string + “.xml”. The Report Type string identifies the type of discrepancies a report contains. The description of the different report types are detailed later in this section. The time stamp allows the administrator to run the utility multiple times without overwriting the output files. Examples of default names with the default report type are:

- ▶ cmrmvolval20020531123456\_FILENOTFOUND.xml
- ▶ cmrmvolval20020531123456\_SIZE\_MISMATCH.xml

There are two default discrepancy reports:

#### **File not found**

Entries are added to the FILENOTFOUND report if an object is in the Resource Manager database, but it is not found on the volume recorded in the database. A file is considered “not found” if the volumes device manager either reports that the file does not exist, or reports that the file has a zero file size when the size in the database is non-zero. The report contains the object information from the Resource Manager database.

**Size mismatch**

Entries are added to the SIZEMISMATCH report if the size of an object in the Resource Manager database does not match the size reported by the device manager. The report contains the object information from the Resource Manager database and the size reported by the device manager.

Archived





## Case study: Retirement Application Processing System

In this chapter, we apply the various technologies, strategies, and options discussed earlier in the redbook to a specific Content Manager solution, the Retirement Application Processing System, which processes retirement application requests.

We describe the business problem, the Content Manager solution that solves the problem, and the backup/recovery, high availability, and disaster recovery practices that are implemented for the system.

This solution consists of multiple sites, multiple machines, and the Sun Solaris platform.

## 7.1 The business problem

A multinational entity has been assigned to set up a new Retirement Application Processing System. The policies for this system are already in place. Starting in 2003, the system needs to handle 20 to 30 million applications each year.

In order to process these applications in a paperless and customer-oriented fashion, a new IT-based system has been developed. It enables automatic processing for most of the requests. In addition, in special incidences and in case of error, the system offers a manual processing capability.

## 7.2 Solution overview

The solution includes much more than just the document management part, which is implemented using Content Manager.

To begin with, because there are many different kind of incoming documents, the solution includes a scanning subcomponent. Incoming paper documents are scanned and Optical Character Recognition (OCR) processing is performed. Metadata is automatically extracted and enriches the documents. Other incoming documents, such as e-mail and faxes, are first printed out to paper and then enter the same cycle of scanning and OCR processing.

Information sent to and received from other authorities, such as the tax authority, or insurance providers, are handled by the messaging component WebSphere MQ messaging. This ensures that no data is lost during transmission and all work is being processed.

Additional services provided by this custom solution include internal communication, data import and export, and a request database. An overview of the components involved is shown in Figure 7-1 on page 221.

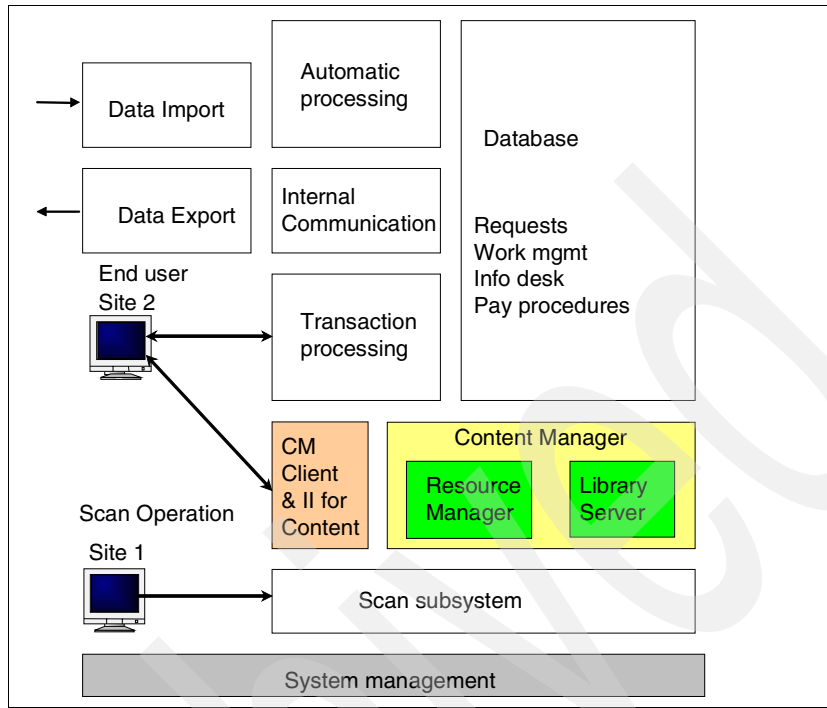


Figure 7-1 Solution overview

Because there are many different components involved, there are different strategies implemented to do high availability and backup and recovery. Section 7.3, “Solution design” on page 223 shows the details for the documents management subcomponent.

First, incoming documents are scanned at scanner workstations. The solution includes more than one scan workstation. The scanning process is managed by an InputAccel server, also responsible for:

- ▶ Image enhancements
- ▶ Validation of extracted metadata
- ▶ OCR
- ▶ Export of data to Content Manager

Figure 7-2 on page 222 shows an overview of the scanning subcomponent, including the network connectivity. Scan clients, scan servers, and Content Manager itself are located in separate LAN segments.

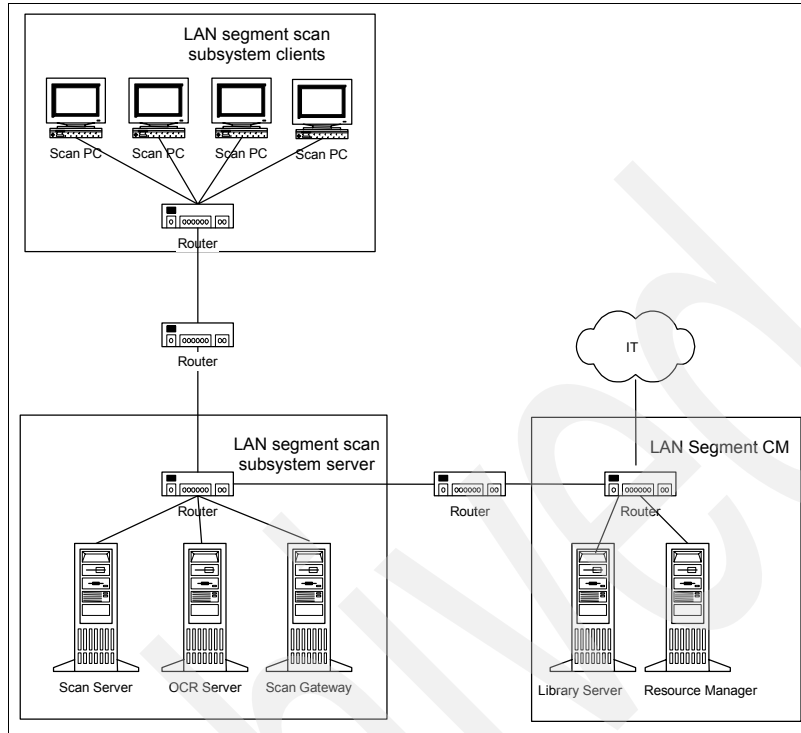


Figure 7-2 LAN architecture Content Manager

In addition to the import of incoming documents, the processing of the requests must be guaranteed. This happens in a mixture of automatic processing and a manual workflow process. The workflow is implemented using both Content Manager document routing and MQ messaging for the exchange of documents with external locations.

A clerk with proper authority can access the system through a Web application, using WebSphere Application Server in a clustered environment. Typically, the clerk gets work assigned based on a Content Manager worklist. The clerk retrieves the document to a Web browser and is capable of responding to the request or adding comments and annotations to it.

Figure 7-3 on page 223 shows the data flow to and from the document management subcomponent. It also contains the expected daily volume of data that is transferred to between these components. On the retrieve side, step 4 is the retrieval of scanned documents and step 5 is the retrieval of annotations.

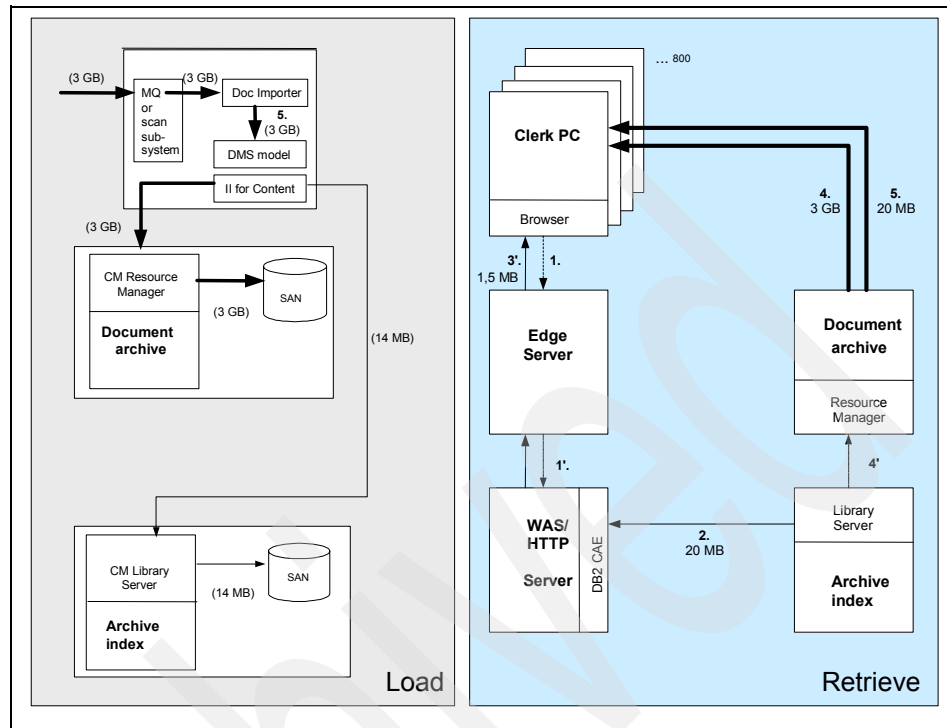


Figure 7-3 Data volume and flow inside the Content Manager component

## 7.3 Solution design

This section discusses system requirements and the solution design to meet these requirements.

### 7.3.1 High availability

The requirement for availability has been defined as 98% on five workdays with 20 hours. During normal business hours, the system should be available for users. During nights and weekends, maintenance down time, such as updating components and creating backups, can occur.

To achieve these specifications, the following provisions have been made during the design:

- IT architecture:
  - Isolation of automatic processing and the dialog system into separate logical and physical systems

- Asynchronous message and transaction-based processing
- Software clustering for WebSphere MQ and WebSphere Application Server components
- WebSphere Application Server cloning
- Hardware clustering by using hot and multidirectional (mutual takeover) standby systems
- Using of automatic load balancing tools
- ▶ Development process:
  - Using a defined development and release process
  - Fixing of known problems by applying patches
- ▶ Service:
  - Integration of the application monitoring into an already in-place monitoring process to detect upcoming concerns early
- ▶ Hardware selection:
  - System availability
 

By using PRIMEPOWER servers together with the operating system Sun Solaris, you get the basic high availability functions. Both offer hardware redundancy such as multiple power supplies and fans; this ensures continuous operation of the system even if one component is failing. Online replacement functionality and hot-plug functionality allow the exchange of defective components while operating the system. When an error is detected in any of the main units, such as processors or memory of disk controllers, the system is rebooted immediately with the faulty component being disabled automatically.
  - System usage
 

The size of the servers building a cluster has been chosen to have a utilization of less than 75% after taking over a failing server. So, the servers are capable of handling an additional 25% workload on top of the 75% usage for a peak situation.

Every machine type was chosen to be upgradable by two additional processors to be able to handle future unknown peak workloads.
  - Data availability
 

By using mirrored hard disks (RAID-1) and RAID-5 for the directly attached disks and inside the SAN for the Symmetrix systems, you get immunity against the failure of a single disk. Online replacement functionality and hot-plug capabilities allow the exchange of faulty disks during operation of the server. By using the synchronous data mirroring capabilities (Symmetrix Remote Data Facility, SRDF) between the two Symmetrix systems, data replication takes place in both directions,

creating additional server workload. It is mainly used for high availability of business-critical data. A fast warm restart after a complete crash of the computer center or a failure of one of the Symmetrix Systems is also available. Frequent scheduled backups are created to further increase data availability.

- Application availability

The high availability of applications and services is achieved by using PRIMECLUSTER HA servers. They can automatically detect system outages or failing components and start appropriate recovery operations. Depending on the type of error, reactions include local recovery actions to complete failover of applications affected by this error.

## **Reliant Monitor Service architecture**

The designed solution uses Reliant Monitor Service architecture for high availability.

### ***Mode of operation***

Reliant Monitor Service (RMS), or PRIMECLUSTER, is a high availability solution of Fujitsu Siemes Computers. With RMS, two or more systems are operated as a high availability cluster.

The high availability cluster needs to comply to the following requirements:

- ▶ Monitoring and control of applications
- ▶ Restart of the applications in case of failover to a secondary system
- ▶ Automatic detection of system outages of component failures
- ▶ Multidirectional or hot-standby failover
- ▶ Take over of resources to the secondary system in case of failover
- ▶ Switch the SRDF mirror in case of a failure of the Symmetrix systems
- ▶ Recovery of clients inside the LAN using IP aliasing
- ▶ Support for disaster recovery

For the productive use of a cluster in this case, PRIMECLUSTER functions used are cluster fail-over management, administration of the cluster-wide data, protection from network errors, and wizards for the easy implementation of the user-specific cluster environment. Figure 7-4 on page 226 describes the architecture of a high availability cluster.

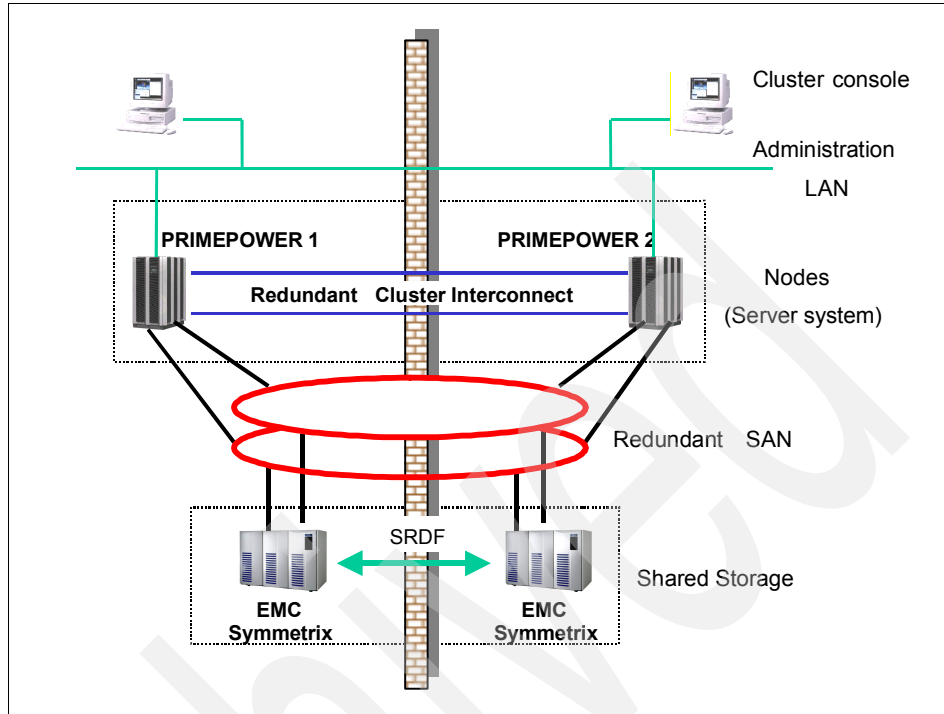


Figure 7-4 High availability cluster layout

### **Connecting the server nodes to the SAN**

For a high availability cluster, all systems need to access a shared physical disk system. This can be a shared disk subsystem or a SAN.

The connection of the nodes to the SAN is done using redundant host bus adaptors. In case of a failure of one of the SAN connectors, you are not losing the whole connectivity. Veritas Volume Manager, for example, contains a dynamic multipath functionality that redistributes the I/O paths in case of connectivity problems. Otherwise, it splits work load and increases throughput by distributing requests to the I/O channels.

### **Heartbeat**

The cluster interconnect is also implemented with redundancy. In case one component is lost, it does not destroy the connectivity within the cluster. The cluster interconnect communicates with the nodes inside the cluster by sending a heartbeat. The heartbeat is responsible for detecting an unavailable node. When a node is sending a message to another node, the other node must respond within a defined time frame. A negative (=no) response does not necessarily mean that an application is down or the access to the storage system is not



working. PRIMECLUSTER uses monitoring agents to watch applications and system components. Every change in its status is reported to the Reliant Monitor Service (RMS).

### ***Reactions to failures***

RMS is configured by rules, defining reactions to the outage of applications or components. Possible reactions include a restart of the application or the failover to a standby system by taking over the resources. In this concrete implementation, the first choice is to restart the application on the original system. Changing to the standby system is the second choice. Scripts have been created for an automatic and manual reaction in case of failures.

### ***Fail-over and fail-back control***

In case of a failover, the failing node will be automatically stopped (power off) by the cluster console. This is mainly done to ensure data consistency in the SAN. After repairing and rebooting the failing node, the failback happens manually by qualified support personnel. This ensures that a failing node can come back online only after support personnel completed all maintenance steps, and this avoids unwanted stepping errors.

## **7.3.2 Backup and recovery**

Basic requirements for backups are to keep at least the last three generations, which is equivalent to 28 days of data. Both systems and databases must be backed up.

### **System backup**

No preparations are necessary for a backup of the custom applications. The application can be backed up while being operative. For restore operations, all process of the application must be manually stopped. There are no scripts prepared for this event.

The data backup inside the SAN happens directly from the application server through the SAN to a tape library. There is no backup server in between. The backup takes place as a serverless backup, meaning that there is no additional workload to the application server or the LAN. High availability backup administration systems keep the backup information and control the operation. The overall system layout for backups is shown in Figure 7-5 on page 228.

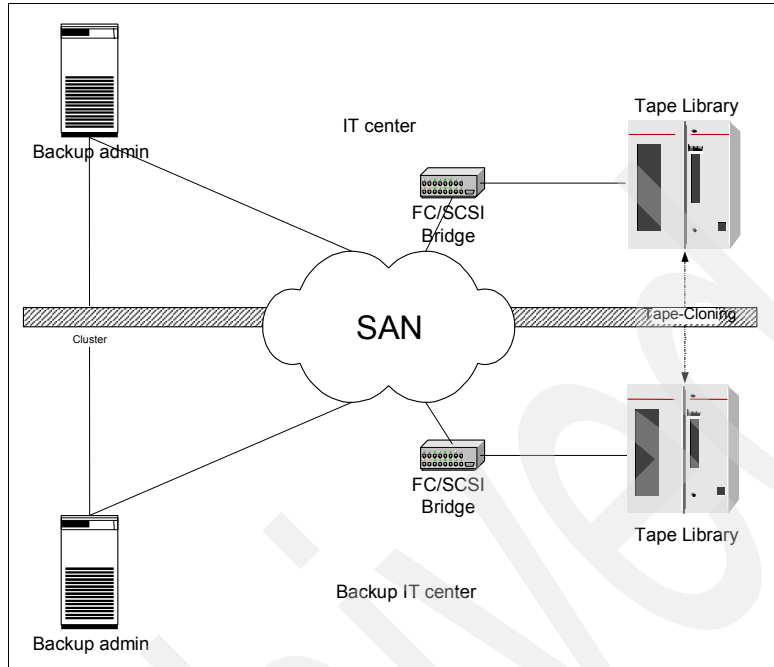


Figure 7-5 Data backup inside the SAN

For most of the systems, a full backup is scheduled weekly. In addition, daily incremental backups are created.

### Database backup

For backup purposes, Veritas backup is used. Advantages of using Veritas are:

- ▶ The backup is done online. The database does not need to be stopped. Users and processes can work with the database during the backup operation.
- ▶ The backup takes place as a snapshot. Opposite to a plain online backup, the database is in backup mode only for a few seconds. During this time, the snapshot is being created. This avoids the creation of large rollforward log files during the backup.
- ▶ The backup is LAN free. The data backups happens inside the SAN. The database server creates a block list of files to be saved. This list is sent to the data mover inside the SAN. The data transfer stays inside the SAN, causing no additional workload to the LAN.

- ▶ The backup takes place serverless at block level. The data is not saved through the database server. The data is sent directly from the disk system (EMC Symetrix) to the tape libraries. The data mover saves the data identified by the block list on I/O level and not at the file system level. Access at the block level is the fastest way to access the data and get a high performance backup. By only saving updated blocks, the data amount is reduced further.
- ▶ Down time of the database in case of failure is determined by the time needed to recover the database. The bigger the amount of changed data since the last full backup, the longer is the time needed to recover the database. By creating many incremental backups during operation, the expected amount required of rollforward recovery after a crash is limited.

A full backup of the databases is being created weekly. In addition, incremental backups are created daily. Log files for rollforward recovery are also saved on a daily basis.

Before the weekly full backup, and anytime in between, the database needs to be stopped, and the complete custom application must be stopped. If the application is not stopped prior to the database shutdown, the custom application remains functional. Thousands of log entries will be created, and over time, queues will get full. Those exceptions will be reported by the monitoring component.

### 7.3.3 Geographic distribution

The components explained in the 7.3.1, “High availability” on page 223 are distributed in two distinct fire protection areas. Mainly, the two redundant servers and the associated storage devices split into these areas. There is no further geographic distribution requirement currently mandated.



## **Case study IBM ECM solution: Personnel Records System**

In this chapter, we apply the various technologies, strategies, and options discussed earlier in this redbook to a specific IBM DB2 Content Manager and e-Records (ECM) solution with a Personnel Records System that manages personnel records for an organization.

We describe the business problem, the Content Manager solution that solves the problem, and the backup/recovery, high availability, and disaster recovery practices that are implemented for the system.

This solution consists of multiple sites, multiple machines, and multiplatform environment involving both AIX and Windows.

## 8.1 The business problem

A multinational entity has a Personnel Records System that manages multimillion personnel records and is accessed by over 1,000 authorized users. Currently, this system, along with other enterprise applications, does not support business content. The existing environment does not promote information distribution and sharing; it does not meet today's e-business demands.

## 8.2 Solution definition and benefits

IBM DB2 Content Manager technology supports the modernization of the Personnel Records System and provides an enterprise-wide process to allow access to business-critical applications. Implementation of the Content Manager and e-Records (ECM) solution provides substantial benefits over the existing system. The new solution provides metadata and full text search of content, including version control, and enables Web browser access. Content Manager assets and its infrastructure strategy allows the organization to provide access to the existing system from all over the world through a single interface, round-the-clock global presence, and support through the Web.

## 8.3 Scope and complexity

The IBM ECM solution, the Personnel Records System, supports automation of personnel records and serves as the official repository of electronic records. The solution provides the capability to create, manage, transfer, retire, and archive electronic records. The various component parts of the electronic records and Content Manager system is integrated into a virtually seamless system accessible from the desktop, and it supports Web access using standard Web browsers. Document imaging, data storage and retrieval, and personnel processing capabilities are provided as well.

The system is also integrated with the organization's portal in a seamless manner, enabling its personnel, including those in different countries, access to their personal information through a standard Web browser.

The Personnel Records System supports up to 1,000 process users and 500,000 general users at a rate of 400 users per hour. Documents to be stored include the current documents found in system (about 20 terabytes of data) and new incoming documents at a rate of 100,000 new pages per day. The system provides the capability to view personnel records by record category (for example, selection folder and correspondence).

User access is based on functional groupings of data and authorized user privileges. Users can access their authorized data and functions from any workstation linked to the network. The system has the ability to establish access policy and grant access requests.

The workstation logon process requires no more than 30 seconds (usually 15 seconds within LAN), and logoff procedures are completed within 20 seconds per solution configuration. Response to data entered at a workstation occurs within 1-3 seconds, or the time needed to refresh or regenerate, or both, a monitor screen.

The following requirements are built-in to the solution to implement backup/recovery, high availability, and disaster recovery. They include backup and recovery requirements, geographical fail-over requirements, availability requirements, and support requirements.

### **Backup and recovery requirements**

The backup and recovery requirements include:

- ▶ Provide access restriction capability to properly authorized personnel for the specification of backup/audit options.
- ▶ Perform full backups every week.
- ▶ Perform incremental backups every day.
- ▶ Rebuild forward from any incremental backup copy, using the backup copy and all subsequent audit trails in order to restore the operating system, database, and document repository within 12 hours to the point of the last incremental backup.
- ▶ Within four hours of failure, provide recovery to the point of the last full backup of the complete configuration, including operating system, database, and file structure of any of the production or test servers following a physical failure of the hardware or corruption of the operating system configuration.

### **Geographical fail-over requirements**

The geographical fail-over requirements include:

- ▶ Any data or documents that are entered into the document repository at site A will be copied asynchronously to the remote site within 10 minutes.
- ▶ In the event of a catastrophic failure of the primary site, the system will fail over to the remote site within one hour.
- ▶ In the event of a failure of the primary site, the number of scanned documents that must be reprocessed by the users at the remote site will not exceed 10,000.

## **Availability requirements**

The availability requirements include:

- ▶ System availability, performance, and uptime should not be negatively impacted for the purpose of software upgrades or installations, or both.
- ▶ The repository system will be available during the hours of 7 a.m. to 12 a.m. Monday through Friday. Down time should not exceed four hours per month.
- ▶ In the event of an outage, the subnet will be restored and be operational in one hour or less, with no more than one hour of lost data.
- ▶ Storage statistics, produced on demand, should provide a detailed accounting of the amount of storage consumed by the processes, data, and records.
- ▶ System management functions should notify system operators of the need for corrective action in the event of critically low storage space.

## **Systems errors**

In case of persistent system errors, the following should happen:

- ▶ Fatal system errors should result in an orderly shutdown of the server application.
- ▶ System error messages should include a text description of the error, the operating system error code (if applicable), the application detecting the error condition, and a date and time stamp.

## **Support requirements**

Support requirements include:

- ▶ Perform user notification for unplanned service outages.
- ▶ In the case of an emergency requirement to shut down a service, provide notification prior to the shutdown.
- ▶ Provide “return to service” status prior to system restore.
- ▶ Receive requests for assistance by phone, e-mail, or fax.

# **8.4 Solution architecture**

Figure 8-1 on page 235 illustrates the IBM ECM solution for the Personnel Records System, including hardware, software, network, personnel resources, and geographically dispersed backup site to implement the disaster recovery strategy (through failover to redundant configuration). Site A is the main production and development site, and Site B is the backup site.



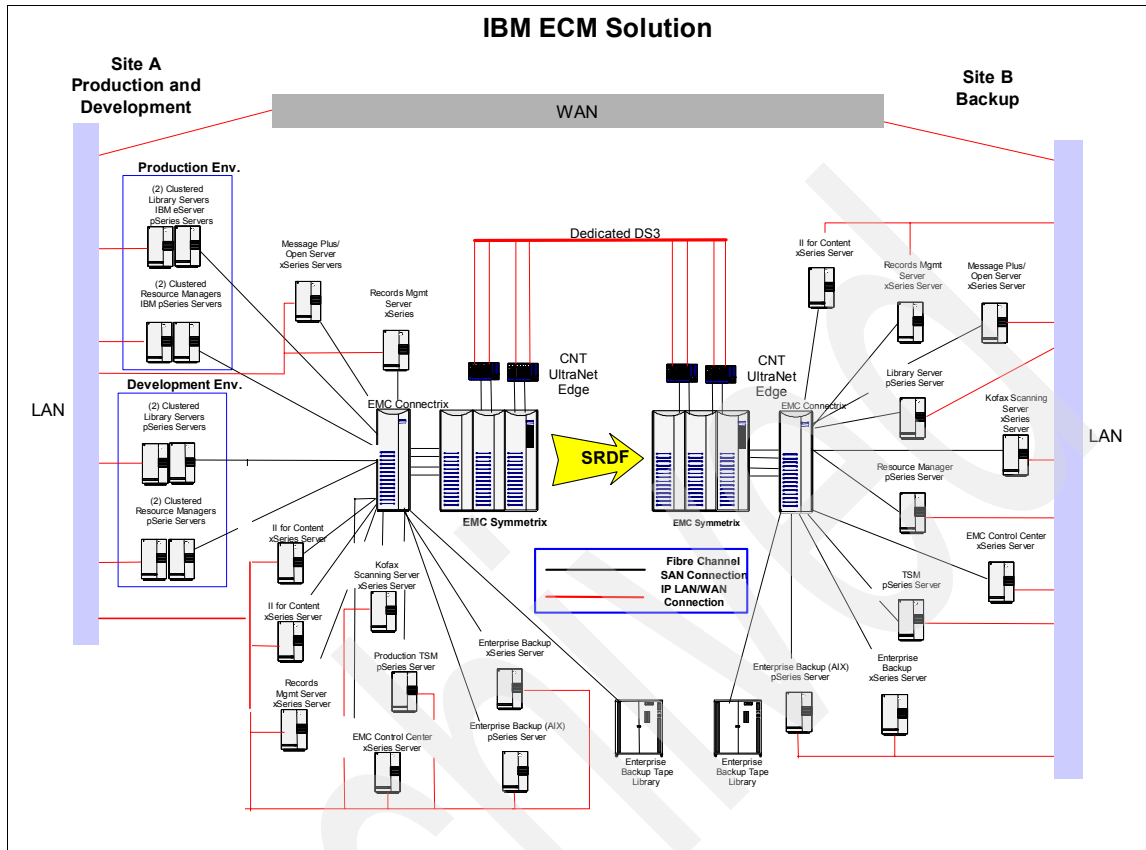


Figure 8-1 IBM ECM solution with dual site DR configuration using SAN and SRDF technology

This configuration integrates IBM @server pSeries and xSeries machines with EMC Symmetrix 8830 and Connectrix SAN equipment connected to the regional fail-over site through CNT UltraNet Edge switches and a dedicated DS3 IP network connection. Fibre Channel is used for the connections to the SAN equipment within each site.

## 8.5 Application and data protection strategy

In this solution, three levels of application, data protection, and recovery are integrated in the design.

### 8.5.1 Disaster recovery strategy

This solution includes three levels of system failover, which provide the functions required to meet the organization's requirements for data protection, high availability, and disaster recovery:

- ▶ The data protection level requirement is addressed by providing incremental and full data backups described in the "Backup and recovery requirements" on page 233.
- ▶ The high availability level requirements is met through design-built redundancy of all components and basic clustering of production systems.
- ▶ The disaster recovery level requirement is achieved with the implementation of a geographical fail-over capability coupled with SAN and data replication technology.

To accomplish this, a replica of the Site A (Central Library) is configured and maintained operational in Site B (Regional Library). In case of application failure, logging out from the Central Library and logging in to the Regional Library is required to fulfill manual geographical failover to the Regional Library. This is accomplished within minutes.

### 8.5.2 Backup and restore

System backup is enacted through a dedicated Veritas software backing up the system configuration to an LTO tape backup system. The Veritas server can be scheduled to perform full systems backups and incremental backups.

By leveraging the EMC Timefinder, BCV capabilities, and Veritas software, a full backup of the operating system, file structure, database, and images can be performed weekly with little to no interruption to the continuous operation of the production system.

An incremental backup of modified data and images (operating system, file structure, and database) since the last full backup can be performed daily through a dedicated Tivoli Storage Manager server without affecting the continuous operation of the production system.

### 8.5.3 System availability strategy

Enhanced availability of the solution is supported through redundancy of functionality and software and hardware components built-in the design of the solution.

## Hardware reliability and availability

The platform layer addresses the basic quality and reliability of a server. All availability levels evolve from this foundation: A platform that experiences a high number of faults has a negative impact on all availability levels. Although it is not possible to design a system that never has a hardware fault, servers can be designed with standard and optional redundant components that provide protection against certain failures. Because completely redundant, fault-tolerant systems can be prohibitively expensive, the objective is to design cost-effective systems that also incorporate redundancy efficiently.

Incorporating high availability and redundant components into the server protects the server from certain faults, but it does not remove all points of failure. If the server cannot access data residing on a storage device, a user will experience down time. Because data is one of the most valuable assets that a business owns, it is critical to apply platform availability attributes to external storage devices. The primary concern of the data layer is to protect physical and logical data residing on storage devices internally or externally attached to the server platform.

Advanced xSeries x-architecture and pSeries Intelligent Service Processor, high-availability, manageability, and serviceability features help diagnose problems quickly, even from remote locations. This includes the following technology:

- ▶ Advanced Chipkill™ ECC memory corrects two-, three-, and four-bit memory errors.
- ▶ Active PCI slots features hot-add and hot-swap adapters.
- ▶ Hot-swap drive bays, fans, and power supplies allow replacing components without powering down the server.
- ▶ Predictive Failure Analysis® (PFA) on processors, voltage regulator modules (VRMs), memory, fans, power supply, and HDD options warn of problems before they occur.
- ▶ Two hot-swap power supplies and up to two additional 250-watt power supplies are optionally available for redundancy.
- ▶ Integrated Advanced System Management Processor supports remote POST, setup, and diagnostics, so the service technician, or Support Center, can put the xSeries 250 back online quickly.

## Data availability

Data availability adds to the platform availability characteristics. It is designed to protect data residing on storage devices attached internally or externally to the server platform, with the level of protection increasing through advancements in storage technology (ATA, SCSI, Fibre Channel). Like platform availability, the

data availability layer introduces features designed to help eliminate single points of failure so that data will be available to users, even if the storage device is in a degraded state (redundant fans, power supplies, and controllers). In order to provide continuous access to data, information must be available if any of the following software or hardware components fail or change: file, directory structure, logical disk volume, physical hard disk drive, host bus adapter or controller, enclosure management module, data bus, or pathway to data.

### **Software availability**

Although redundancy at the platform and data infrastructure layers helps reduce outages, it does not offer protection in the event of a software or complete system failure. Organizations requiring increased availability above the platform and data layer can deploy system-level redundancy to provide application availability. This portion of the continuum focuses on providing a mechanism to help ensure the health of the application to users.

While the platform and data layers focus on hardware redundancy within individual products, the application layer focuses on using software tools for monitoring applications and on coupling systems together to provide improved application availability. The software tools, or middleware, monitor the health of an application by detecting application hangs, data accessibility, networking dependencies, and full system failures and then performing the necessary actions to route around a failure or to get the application back in service.

Coupling systems together for redundancy is known as clustering. Clustering can take on multiple forms, but the most common forms help provide system availability should a server go down.

In this solution, the software redundant configuration, including a standby production Content Manager Library Server, Content Manager Resource Manager, Records Manager server, and dual Information Integrator for Content (formerly known as EIP) servers, provides enhanced availability capability. EMC Timefinder and BCV capabilities satisfy the high availability requirements using flash copy and mirroring technology that delivers automated fail-over mechanisms.

### **Geographical availability**

The first three layers of availability serve the majority of critical application environments. There are applications and data so critical in nature that they need further protection. Load balancing and high availability cluster solutions improve availability at a particular site, but system down time can be caused by a site-wide natural disaster (such as a fire, earthquake, and flood) or man-made disaster.

Protection for applications and data at this level requires thoughtful planning, considerations, and, sometimes, special products for replicating information and systems at a remote site.

The most cost effective method of protecting against a site disaster is to perform routine backups and archive data at a different location. This can enable the restoration data and systems manually either at the damaged site or at another site. These processes do not require any unique hardware, but are highly dependent on best practices to ensure the data is as up-to-date as possible.

In many instances, the best practices for backup do not provide enough protection should a disaster occur during normal business hours, when data loss can be substantial despite a recent backup.

To account for this type of event, the IBM ECM solution replicates data from Site A to Site B. Solution flexibility is critical at this layer, because each environment has unique requirements. These site-to-site replication schemes allow for either asynchronous or synchronous updates that simultaneously update data at both locations. For additional availability, servers are deployed at a secondary site to handle the primary sites' application workload when it is needed.

### **Network configuration**

In this case, as in most cases, a careful assessment of existing network capabilities and an evaluation of new network configurations, required for disaster recovery, were conducted at the time of designing the overall architecture of this solution.

Fibre optic cables, redundant switches, and two dedicated DS3 lines are included in the design to provide sufficient redundancy and throughput within the sites and between the two sites. See Figure 8-2 on page 240 and Figure 8-3 on page 241.

The following configurations are in place for this IBM ECM solution:

- ▶ An Ethernet 10/100 connection is available for connectivity to the existing wiring closets. TCP/IP is the only protocol on the LAN and WAN.
- ▶ There are pSeries servers and xSeries servers in the Site A Data Center requiring GB Ethernet ports and 10/100 Ethernet ports.
- ▶ There are two VLANS configured: Production and Development.

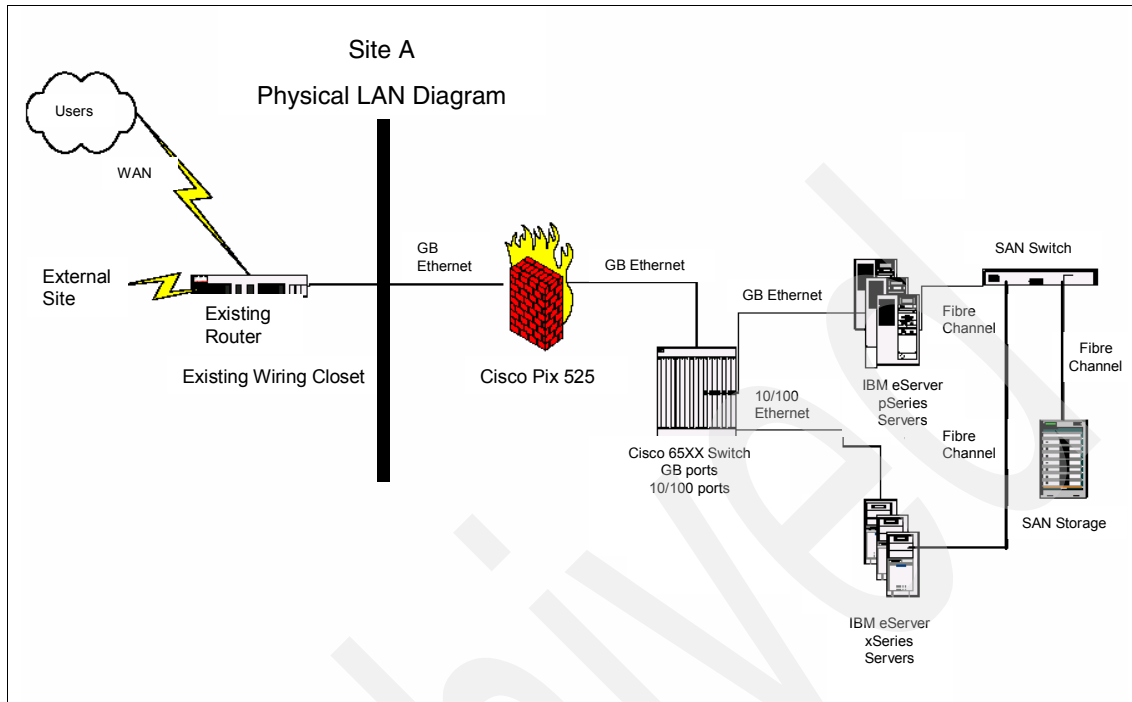


Figure 8-2 Site A network topology

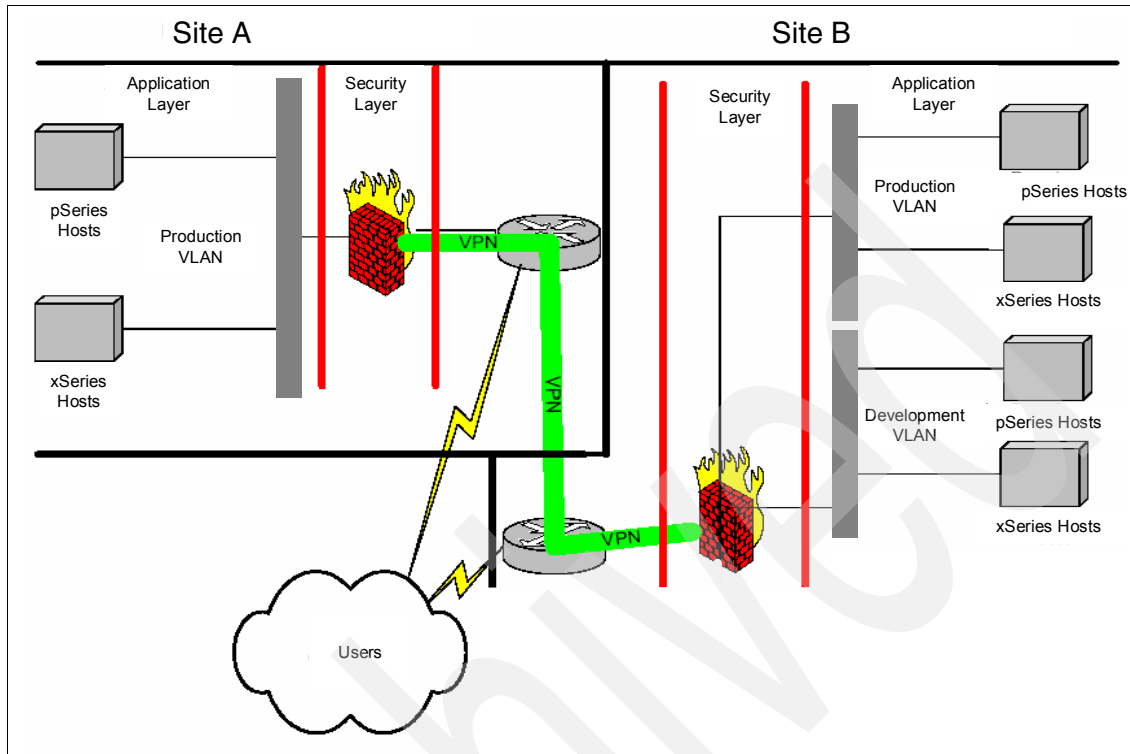


Figure 8-3 Site A and Site B network topology

- ▶ There are pSeries servers and xSeries servers in the Site B Data Center (similar to that of Site A) requiring GB Ethernet ports and 10/100 Ethernet ports.
- ▶ A site-to-site VPN is configured between the Site B PIX firewall and the Site A PIX firewall. This VPN does not require client authentication.
- ▶ Intrusion detection is required at both locations. This service is supplied by the IDS card in each Cisco 65xx switch.
- ▶ Routing is available at both sites.
- ▶ Failover between sites is performed manually.
- ▶ Switches require redundant power supplies.

### Training, services, and system maintenance

The human resources aspect of disaster recovery planning is of utmost importance because it is the qualified IT personnel who will implement the disaster recovery plan procedures to ensure a high quality data protection service. The IBM ECM solution includes provisions for IT personnel training and

the necessary resources to prepare people and systems for disaster recovery including a development and test environment. In addition, technical support, maintenance, and consulting services are included in the solution to facilitate the system stand-up process and the integration with existing systems. These include:

- ▶ Infrastructure consulting services for migration and integration of different environments, including mission-critical e-business systems.
- ▶ Technology Exploration Centers provide a secure, comprehensive lab for tuning and proof of concept for Content Manager and storage systems.
- ▶ IBM Learning Services provide customer training for servers, storage, clustering, systems management software, and information management software.
- ▶ Enterprise hardware implementation ranging from on-site SAN design to server and rack installation.
- ▶ Technical support for high availability.
- ▶ Business continuity and disaster recovery services.

#### **8.5.4 Full recovery**

This section refers to the recovery of the configuration of any of the machines following a physical catastrophic failure of the hardware, or corruption of the operating system configuration. In this respect, it is different from the application availability function.

The operating system, file structure, and database (excluding images previously backed up) will be recoverable within four hours of failure, to the point of the last full backup (no more than one week old). If stored images are required to be recovered, recovery time will vary depending on the number and volume of objects that have been stored.

#### **8.5.5 Incremental recovery**

The operating system, file structure, and database will be recoverable within approximately 12 hours of failure to the point of the last incremental backup (no more than 24 hours old). The exception to this will be when a full backup is preferable or required due to corruption beyond the latest incremental backups.

##### **Images**

Images will be recoverable as part of any full or incremental backup. In addition, individual images, full member records, or groups of related or unrelated images



can be recovered separately from a full or incremental backup previously described.

Archived



## Sample scripts and programs

This appendix provides the details about the programs we used during the high availability and backup and recovery test scenarios. We provide the source code and a detailed explanation for the following programs:

- ▶ Library Server startup and shutdown scripts for HACMP
- ▶ Sample Java API program used as a custom client

## A.1 HACMP Library Server startup and shutdown scripts

As specified in 5.2.8, “Defining resource groups” on page 160, we need to provide user-defined startup and shutdown scripts for the application we want to make highly available in order to include it in the HACMP resource group. These scripts include the automated steps specific to the application that HACMP needs to perform when moving the resource group from one node to the other.

**Note:** These scripts must be tailored for each particular installation. Your HACMP and Content Manager infrastructure specialists need to spend time working together on these scripts as a critical part of the Content Manager HACMP implementation. The scripts we used in our test environment are provided only as a *reference*.

### A.1.1 Library Server startup script

The Library Server startup script, as shown in Example A-1, is called by the HACMP daemons to start the Library Server on a specific node after all the rest of the resources, such as IP address and disks, are already available.

In the case of a failure in the primary node, the secondary node will take over the shared disks that were being used by the former and try to start the application using this startup script. In this case, it is most probable that the Library Server will not be able to shut down gracefully. For this reason, it is very important to make the script capable of starting a Library Server that was previously shut down without control. It must be able to deal with leftovers such as open shared resources or temporary files. In fact, the most difficult part of the HACMP configuration for a particular component consists of identifying all the possible states of a failing application and creating this automatic recovery and startup script.

*Example: A-1 HACMP Library Server startup script*

---

```
#!/usr/bin/ksh
LOG=/tmp/start_as_cm`date +%m%d%y_%I%M`~
DB2NODES=/home/db2inst1/sqllib/db2nodes.cfg
DB2NODETEMP=/tmp/db2nodes.cfg

{
echo "Removing sems./shared memory"
su - db2inst1 -c /home/db2inst1/sqllib/bin/ipclean
/usr/bin/rm /tmp/DB2INST1*

echo "Fixing hostname in DB2 instance configuration"
```

```

HOST=`/bin/hostname | cut -d'.' -f1`
CUR_HOST=`su - db2inst1 -c cat $DB2NODES | cut -d' ' -f2`

if ["$HOST" != "$CUR_HOST"] ; then
 echo "Changing db2nodes.cfg"
 sed -e s_${CUR_HOST}_${HOST}_g $DB2NODES > $DB2NODETEMP
 cp $DB2NODETEMP $DB2NODES
 chown db2inst1:db2iadm1 $DB2NODES
fi

echo "Starting database manager"
su - db2inst1 -c db2start

echo "Starting library server monitor"
/etc/rc.cmlsproc
} >$LOG 2>&1

exit 0

```

---

The first thing you will note is that the script has almost no error checking. The error checking should be added in a real-life implementation, allowing the script to react to different scenarios. In our case, we kept it as simple as possible for clarity. Note that the script must exit with return code 0 for the failover to succeed.

The first part of the script is variable setup for easy maintenance. After that, we perform resource cleanup in case the Library Server shutdown was not graceful. In our environment, we only need to take care of the DB2 instance startup after a system crash. After doing some research, we decided that it would be enough if we run the DB2 resources cleanup program and erased any temporary file that starts with the name of the DB2 instance, as follows:

```

/home/db2inst1/sqllib/bin/ipclean
rm /tmp/DB2INST1*

```

You should test your startup scripts in your environment and make sure you clean up everything in your environment at this stage.

Next, we need to adjust the host name configuration for DB2 to run in the node of the cluster that is trying to start the Library Server. To do this, we need to have the correct host name in the db2nodes.cfg file for the DB2 instance. In our environment, we are not changing the host name of the machine when the takeover occurs. Therefore, we need to correct this file to contain the host name of the new owner node.

Finally, we start the DB2 database manager for the instance that holds the Library Server database and also the Library Server monitor.

We called this script `startcm_as.sh` and placed it in `/home/icmadmin` in both the primary and secondary nodes of the cluster.

### A.1.2 Library Server shutdown script

The Library Server shutdown script, as shown in Example A-2, is called by HACMP when you bring down the Library Server resource group for maintenance or for migration from one node to another in the cluster.

The goal of this script is to shut down as gracefully as possible all the applications that conform to this part of the resource group. In the case of the Library Server, this includes the DB2 database manager for the Library Server database instance and the Library Server monitor.

*Example: A-2 HACMP Library Server shutdown script*

---

```
#!/usr/bin/ksh
LOG=/tmp/stop_as_cm`date +%m%d%y_%I%M`~

{
echo "Stopping library server monitor"
/etc/rc.cm1sproc -shutdown

echo "Disconnecting DB2 applications"
su - db2inst1 -c "db2 force applications all"
sleep 10

echo "Stopping DB2"
su - db2inst1 -c db2stop
RC=$?
if ["$RC" -eq 0] ; then
echo "DB2 Stopped"
else
echo "Forcing DB2 shutdown"
su - db2inst1 -c db2stop force
fi
} >$LOG 2>$1

exit 0
```

---

Again, this script almost does not contain any error checking for simplicity and clearness purposes.

In this script, we first shut down the Library Server monitor using the scripts provided with Content Manager. Then, we disconnect any other applications that might be accessing the database in this instance such as the Resource Manager services. Finally, we shut down the DB2 database manager.

Note that we need an exit status of 0 for successful resource group shutdown. We also must make sure that DB2 stops even if we need to force connections down or kill processes. This will leave the system in a better condition than when the system is halted at a lower level to allow resource group migration to another node.

## A.2 Sample custom API program

This is the simple command-line Java program, as shown in Example A-3, we used as a client to run the backup tests in Chapter 3, “Practical backup and recovery procedures” on page 61 and the fail-over tests in Chapter 5, “Practical procedures for high availability” on page 121.

In the backup and restore scenarios, the idea behind this utility is to simulate user activity importing new items to the system during online backup. This way, we could see how big the inconsistency was between the different components, such as databases and file systems, after a point-in-time restore.

For the fail-over test, we wanted to observe the reaction of the Content Manager API when the connection to either the Library Server, Resource Manager, or both, is lost due to a failure and what are the steps needed to continue normal operation after the failover occurs.

*Example: A-3 Sample custom API program*

---

```
// Import required CM API packages
import com.ibm.mm.sdk.common.*;
import com.ibm.mm.sdk.server.*;
// We need the following package to handle DB2 exceptions
import COM.ibm.db2.jdbc.*;

// Basic Java functions imports
import java.io.*;
import java.util.*;

public class JImportStuffForBackupTest {

 // A few variables to make changes a little bit easier
 static String database = "icmhadb";
 static String userid = "icmadmin";
 static String pw = "password";
 static String itemtype = "TSM_01_Repl";
 static String AttrName = "Description";
 static String FileName = "c:\\CMDaemon\\Sample.jpg";
 static DKDatastoreICM ds;
```

```

static intAmount = 30; // Amount of images to import
static int Interval = 2; // Interval (in seconds) between imports

public static void main(String[] args) {

 int i=0;
 boolean bEnd=false;

 // Connect to CM datastore
 try {
 ds = new DKDatastoreICM();
 ds.connect(database, userid, pw, "");
 } catch(Exception e) {
 System.out.println("Exception during initialization. " + e.toString());
 e.printStackTrace(System.out);
 return;
 }

 // Main loop to import images
 while(!bEnd) {
 try {
 for(;i<Amount;i++) {

 // Create DDO, add XDO with content and store into datastore
 DKDDO ddo = ds.createDDO(itemtype, DKConstant.DK_CM_DOCUMENT);
 ddo.setData(ddo.dataId(DKConstant.DK_CM_NAMESPACE_ATTR,AttrName), "" + i);
 DKLOBICM base = (DKLOBICM) ds.createDDO("ICMBASE", DKConstantICM.DK_ICM_SEMANT
IC_TYPE_BASE);
 base.setMimeType("image/jpeg");
 base.setContentFromClientFile(FileName);
 DKParts dkParts = (DKParts) ddo.getData(ddo.dataId(DKConstant.DK_CM_NAMESPACE_
ATTR,DKConstant.DK_CM_DKPARTS));
 dkParts.addElement(base);
 ddo.add();

 System.out.println("Store doc # " + i);

 Thread.sleep(Interval*1000);
 }

 // Disconnect from datastore
 ds.disconnect();
 ds.destroy();

 bEnd = true;

 // In case of DB2 or CM exception, assume system failure and try to reconnect
 } catch(DB2Exception e) {

```



```

 System.out.println("Disconnected from datastore. " + e.toString());
 bEnd = reconnect();

 } catch(DKException e) {

 System.out.println("CM Exception recieved. " + e.toString());
 bEnd = reconnect();

 } catch(Exception e) {

 System.out.println("Unhandled Exception. " + e.toString());
 e.printStackTrace(System.out);

 bEnd = true;
 }
}

// Reconnection routine
private static boolean reconnect() {

 // Wait an interval, destroy the current connection and try to create a new one
 try {
 Thread.sleep(Interval*1000);
 ds = null;
 ds = new DKDatastoreICM();
 ds.connect(database, userid, pw, "");
 return false;

 // In case a new CM exception is recieved, try to reconnect again
 } catch(DKException e) {

 System.out.println("CM Exception recieved. " + e.toString());
 e.printStackTrace(System.out);
 return reconnect();

 } catch(Exception e) {

 System.out.println("Unhandled Exception. " + e.toString());
 e.printStackTrace(System.out);
 return true;
 }
}
}

```

---

Note that we need to destroy any reference to the current datastore object and create a new one to retry the connection when a Library Server failure occurs. We unsuccessfully tried disconnecting or destroying the connection to the

datastore before using this approach. You do not even need to disconnect to the datastore if a Resource Manager failure occurs. Just retry the operation a certain number of times until you get a positive response from the system.

# Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

## IBM Redbooks

For information about ordering these publications, see “How to get IBM Redbooks” on page 254. Note that some of the documents referenced here may be available in softcopy only.

- ▶ *Performance Tuning for Content Manager*, SG24-6949
- ▶ *Content Manager Implementation and Migration Cookbook*, SG24-7051
- ▶ *eClient 101 Customization and Integration*, SG24-6964
- ▶ *IBM WebSphere V5.0 Performance, Scalability, and High Availability: WebSphere Handbook Series*, SG24-6198
- ▶ *DB2 Warehouse Management: High Availability and Problem Determination Guide*, SG24-6544
- ▶ *Best Practices for High-Volume Web Sites*, SG24-6562
- ▶ *IBM TotalStorage Solutions for Disaster Recovery*, SG24-6547
- ▶ *Disaster Recovery Strategies with Tivoli Storage Management*, SG24-6844
- ▶ *A Practical Guide to DB2 UDB Data Replication V8*, SG24-6828
- ▶ *DB2 UDB Exploitation of the Windows Environment*, SG24-6893
- ▶ *Backing Up DB2 Using Tivoli Storage Manager*, SG24-6247
- ▶ *Content Manager VideoCharger Installation and Integration for Multiplatforms*, SG24-6410

## Other publications

These publications are also relevant as further information sources:

- ▶ *IBM Content Manager for Multiplatforms: Planning and Installing Your Content Management System*, GC27-1332
- ▶ *IBM DB2 Content Manager for Multiplatforms: System Administration Guide*, SC27-1335

- ▶ *Content Manager Application Programming Reference*, SC27-1348
- ▶ *HACMP for AIX: Concepts and Facility Guide, Version 4.5*, SC23-4276
- ▶ *HACMP for AIX: Installation Guide, Version 4.5*, SC23-4278
- ▶ *IBM Tivoli Storage Manager for Windows: Administrator's Guide*, GC32-0782
- ▶ *IBM Tivoli Storage Manager for Windows: Backup-Archive Clients Installation and User's Guide*, GC32-0788
- ▶ *IBM Tivoli Storage Manager for UNIX: Backup-Archive Clients Installation and User's Guide*, GC32-0789
- ▶ *IBM DB2 Universal Database Enterprise Edition for AIX and HACMP/ES*, TR-74.171

## Online resources

These Web sites and URLs are also relevant as further information sources:

- ▶ *IBM Content Manager V8.2 Performance Tuning Guide*  
<http://www.ibm.com/software/data/cm/cmgr/mp/support.html>
- ▶ Content Manager main site  
<http://www.ibm.com/software/data/cm>
- ▶ Tivoli Storage Management software main site  
<http://www.ibm.com/software/tivoli/products/storage-mgr>

## How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

[ibm.com/redbooks](http://ibm.com/redbooks)

## Help from IBM

IBM Support and downloads

[ibm.com/support](http://ibm.com/support)

IBM Global Services

[ibm.com/services](http://ibm.com/services)

# Index

## Symbols

.rhosts 151  
/etc/hosts 150

## Numerics

80/20 rule 104  
9/11 199

## A

access modules 52–53, 63  
access path 8  
accidents 9  
active/active configuration 118  
active/active mode 105  
active/standby configuration 118  
active/standby mode 105  
adapter 151  
API 3  
application clustering 98, 104, 110, 113  
application server 137  
Apply program 115  
architecture 62  
    Content Manager 2  
archival logging 15, 19–20, 22–24, 29  
archive copy group 21  
archived logging 79  
aristotle 123, 126, 130, 142  
Asynchronous Recovery tools 37  
Asynchronous Recovery utilities 38–39  
Asynchronous Recovery utility 85, 208  
ATA 237  
AUDIT VOLUME 89  
authority 222  
availability 92–93, 95  
    measuring 99–100  
availability levels 97  
availability matrix 95

## B

backup 13, 21–25, 27, 31, 39, 44, 66–67, 75,  
83–84, 86, 90, 93, 227, 242  
    cold backup 13

    hot backup 14  
    Library Server 14  
    offline 16, 68  
    online 82  
    restore 81  
    summary of components 65  
    Tivoli Storage Manager 79, 87  
    Tivoli Storage Manager system 88  
backup and recovery 5  
BACKUP DEVCONF 89  
backup image 25  
backup images 30  
backup node 126, 129, 147  
backup pending status 20  
backup requirements 8  
backup tiers 12  
BACKUP VOLHIST 89  
backup-archive client 40–41, 43–44  
backupConfig 59  
bandwidth 37  
BCP 196–197  
boot adapter 151  
BPCP 196  
business continuity 242  
    trend 200  
business continuity plan 196–197  
business process contingency plan 196

## C

C++ compiler 21  
cache objects 35  
Capture process 115  
cascading without fallback  
    HACMP strategy 186  
central scheduler 44  
chdev 144  
chmod 136  
chown 136  
chvg 146  
circular logging 15–16, 22–24, 73, 79  
Cisco 241  
cl\_resgrp 167  
claddgrp 162

- claddnode 158
- claddserv 163
- clclare 166
- clfindres 167
- Client for Windows 49, 68
- client node 179
- client-server software 25
- clmodnetwork 153–154
- clnodename 153
- clstart 166
- clstop 166
- clstrmgr 166
- cluster 125, 226
- cluster daemon 166
- cluster ID
  - set 152
- cluster node
  - add 153
- cluster resource
  - synchronize 165–166
- cluster service
  - stop 166
- cluster topology
  - synchronize 160
- clustering 98, 118, 238
- cold backup 13
- collection 174
- collections 37
  - enable for replication 173
- compiler 21
- configuration
  - multiple node 86
- configuration files 52, 64
- configuration parameters 44
- content
  - managing 62
- Content Manager
  - component overview 62
  - configuration directory 141
  - introduction 2
  - restart 80
- Content Manager client nodes 44
- continuous availability 93
- controls access 62
- copy group configuration 43
- copy groups 73
- Copy Serialization 42
- copy storage pool 88, 90
- crfs 135

## D

- data availability 237
- data redundancy 98
- database
  - Library Server 63
  - reorganize 66
  - Resource Manager 65
  - restore 14
- database logs 57
- database recovery 66
- database replication 113–116
- DB2 137
- DB2 archived log files 21
- DB2 backup image 25
- DB2 database backup 44
- DB2 database manager
  - stop 143
- DB2 databases 19
- DB2 logging 14–15
- DB2 replication options
  - advantages and disadvantages 116
- DB2 user exit 73, 78
- db2adutl 25, 78
- DB2CLASS 73
- DB2CLASS management class 74, 77
- db2cmd 39
- DB2DFTPATH 138
- DB2ENVLIST 141
- DB2EXT schema 67
- DB2EXT.DBDEFAULTS 54
- DB2LIBPATH 141
- DB2LS.properties 149
- db2profile 140–141
- db2stop 143
- db2uext2.c 21
- db2uext2.ctsm 21, 78
- deleted documents 37
- deletion reconciliation utility 39
- delta backup 27
- devhist.out 89
- device class 88
- differential backups 42
- disaster 9
- disaster recovery 5–6, 242
  - strategy 236
  - tiers 205
- disaster recovery plan 196
- disaster recovery planning 196
- disastrous events 198

- disk mirroring 117
- disk pool 108
- disk redundancy 98
- distributed work environments 196
- DNS 107, 115
- document deletion utility 38
- document management 220, 222
- domain name server 107
- down time 92, 96
- drive label 39
- DRP 196–197
- DS3 line 239
- dscenu.txt 76
- dsm.opt 43
- dsm.sys 43
- DSMI\_CONFIG 21, 77–78
- DSMI\_DIR 21, 76, 78
- DSMI\_LOG 21, 77–78
- dual load 119

## E

- earthquakes 9
- eClient 4, 49, 59, 68, 107, 181, 184, 188
- ECM 232
- EIP 238
- enhanced availability 206
- enterprise content 64
  - managing 62
- Enterprise storage disk mirroring 115–116, 119
- Enterprise storage disk mirroring strategies 119
- Enterprise Storage Server 115, 119
- e-Records 232
- ESS 109, 115, 119
- Ethernet 239
- euclid 123, 126, 130, 138, 142–143, 149, 182

## F

- failover 98, 118, 182, 233
  - Library Server and Resource Manager 192
- fail-over platform cluster 105
- fail-over scenario 105
- fail-over settings 177
- Fibre Channel 235, 237
- Fibre optic cable 239
- fires 9
- firewall 96, 107, 241
- floods 9
- FSLog logical volume

- create 132
- full database backup 25
- fuser 143

## G

- generate 45
- geographical availability 238
- group
  - creation 135

## H

- HACMP 108, 125, 127, 207
  - post-config procedures 166
- HACMP configuration 137, 142
- HACMP strategy 186
- HAGEO 119
- hardware clustering 124
- hdisk 126
- heartbeat 226
- high availability 5, 92–93, 95, 97–98, 206
- High Availability Cluster Multi-Processing 125
- high availability component matrix 107
- High Availability Geographic Cluster 119
- high availability options 31
- high availability scenario 122
- high availability solutions 104
- holistic approach 201
- horizontal clustering 104
- hot backup 14
- hot standby 105, 118
- HTTP server 72
  - Resource Manager 65

## I

- I/O controller 8
- icmcreatelsdb.sh 141
- ICMNLSDDB 48
- ICMPLSAP 48
- icmprepenv.sh 210
- icrmr\_asyncr\_logging.xml 210
- icrmr\_validator\_logging.xml 56, 212
- icrmrmdel.bat 39
- icrmrmdel.sh 39
- icmrmlsval.bat 56, 211
- icmrmlsval.sh 56, 211
- icrmrmx.bat 39
- icrmrmx.sh 39

- icrmvolval.bat 211
- icrmvolval.sh 211
- ICMSTRESSOURCEMGR 191
- ICMSTSYSCONTROL 49
- IDM.properties 59
- IDMadmindefaults.properties 59
- import 222
- importvg 145
- incremental backup 26–27, 40–41, 44, 88, 208
- incremental backup strategy 27
- information distribution 232
- Information Integrator for Content 58, 238
- inittab 137
- insourced 201
- installation
  - Content Manager 14
- IP Address Takeover 150
- IP sprayer 96, 107
- IPAT 150
- IP-based adapter
  - add 154–156
- IP-based cluster failover 98
- IP-based network
  - add 153

## J

- JavaServer Pages 4
- JFS 126
- JFS log
  - create 131
- JFS logical volume
  - create 133
- JFSLog 126
- JFSLog volume
  - format 134
- journaled file system 126
- journaled file system log 126
- JSP 4, 59

## L

- LAN Cache 35
- Large File Enabled Journaled File System 134
- LBOSDATA 30
- LDAP integration 127
- Library Server 3, 24, 47–49, 51, 62, 67, 69, 75, 86, 108, 111, 123, 126, 128, 137, 147, 167, 181, 191
  - backup and recovery 14
  - components 63

- configuration files 64
- creation utility 141
- database 63
- failover 182
- fail-over process 183
- log files 64
- LVM layout 130
  - physical config for HA 123
- Library Server database 38, 47, 148
- Library Server monitor 38, 47, 68, 124, 149, 178, 188
  - stop 143
- Library Server Validation Utility 56
- Library Service monitor 48
- life cycle management 2
- load balancer 96, 107
- load balancing 38, 98, 107
- load balancing mechanism 107
- log files 15–16, 19, 21, 29, 63–64, 82, 84
  - primary and secondary 15
  - rollforward recovery 229
- log mirroring 114–116
- log shipping 114–116
- logical host 105
- logical model
  - Content Manager 2, 101
- logical partition 39, 123
- logical tiers
  - Content Manager 102
- logical volume 126
  - create 131
- logical volume manager 125
- LOGPRIMARY 16
- LOGRETAIN 16, 20, 24
- LOGSECOND 16
- LS 126
- lspv 144–145
- LTO tape backup 236
- LV 126, 131, 134
- LVM 125, 142
- LVM layout
  - Library Server 130

## M

- maintenance 26
- management class 73
- matrix
  - availability 96



- MAXTXDURATION 49
- Media Archive Server 36
- media failure 9
- MGT\_CLASS 21, 78
- Microsoft Cluster Service 108
- mid-tier application 107
- mid-tier server 4
- migrate
  - Tivoli Storage Manager 30
- migrator 33, 37–39, 48, 68, 170, 186
- mirrored hard disks 224
- mirroring
  - log 114
- mkgroup 136
- mkiv 132–133
- mkuser 136
- mkvg 131
- mount 135
- mount point 39, 135
- MSCS 108
- Multimedia Archive 30, 32
- mutual takeover 118

## N

- Net Search Extender 54, 63, 67–68
- Net Search Extenders 13
- netstat 155
- netsh.conf 150
- network addresses 125
- NFS 146
- node 105
- non IP-based adapter
  - add 157
  - change 157–158
- non IP-based network
  - add 154
- non-IP cluster failover 98
- non-progressive incremental backup 44
- NSE 13, 63, 67
- NSE indexes 50, 54

## O

- OBJ\_STATUS 31
- object 43
- objects
  - remove 37
- offline 26
- offline backup 16, 22–23, 29, 68, 72, 83

- online backup 22, 24, 51, 82–83
- optical
  - backup 25
- orphan
  - Resource Manager/Library Server reports 214
- outages 93
- outsourced 201
- ownership
  - changing 136

## P

- partition 39
- passwordaccess 45
- pause utility
  - Content Manager 48
- pauseserver.bat 49
- pauseserver.sh 49
- pending status 20
- performance 15, 37, 51, 66, 128
- permission
  - changing 136
- Personnel Records System 232
- PFA 237
- physical disk 15
- physical partition 39
- physical volume 126
- physical volume ID 144
- platform 23
- platform clustering 98, 104, 110
- platform clustering strategy 118
- policy domain
  - backup 73
- policy domain configuration 42
- policy set 74
- postschedulecmd 44
- PPRC 115
- Predictive Failure Analysis 237
- preschedulecmd 44
- preset lower limit 35
- preset upper limit 35
- primary log files 15
- primary node 126
- primary Resource Manager 38, 55
- primary storage pool 88
- protocol 115
- pSeries 123
- purger 33, 35, 48, 68, 170, 186
- PV 126

PVID 144

## Q

QUERY DBVOLUME 89  
QUERY LOGVOLUME 89

## R

RAID 8, 224  
RAID x 98  
rc.cmlsproc 143  
rebind 67  
reconciliation 38  
recovery 5, 10, 31  
    full 242  
    incremental 242  
    Library Server 14  
Recovery Point Objective 199, 203  
Recovery Time Objective 199, 203  
Redbooks Web site 254  
    Contact us xii  
redundancy 238  
redundant switch 239  
regional failover 235  
Reliant Monitor Service architecture 225  
remove objects 37  
reorg 67  
reorganize  
    database 66  
replication 55, 98, 110, 112, 173, 175  
    database 115  
replication options  
    DB2 116  
replication strategy 99  
replicator 32–33, 37, 55, 68, 170, 186  
resource group 126, 161, 164, 167  
    add 162–163  
    defining 160  
Resource Manager 3, 24, 29–31, 37–40, 42–43,  
46, 51, 55, 69, 108, 111, 123–124, 126, 168, 170,  
178, 187, 189, 191, 211  
    components 64  
    database 65  
    failover and fallback 186  
    HTTP server 65  
    replication 55  
    services 65  
    storage area 65  
    volume validation utility 211, 215

Web application 65  
WebSphere Application Server 65  
Resource Manager services 32, 47–48, 186  
Resource Manager/Library Server  
    reports 214  
    validation utility 211  
restart 26  
restarting script  
    Content Manager 80  
restore 22, 82, 84, 87  
    database 14, 19  
restoring  
    Tivoli Storage Manager 81  
resumeserver.bat 49  
resumeserver.sh 49  
Retain Extra Versions 43  
Retain Only Versions 43  
retention 74, 76  
RM 126  
rm 148  
RMLSITEMS 213  
RMOBJECTS 31  
RMS 225, 227  
ROI 197  
rollback 15  
rollforward 14, 29, 84  
rollforward pending 22  
RPO 199, 203  
RTO 199, 203  
RUNSTATS 211

## S

SAN 119, 207, 224, 227, 235  
SAN solution 109  
SAN switch 108  
scheduling facility 44  
SCSI 108, 237  
secondary copy 30  
secondary log files 15  
secondary storage device 40  
services  
    Resource Manager 65  
setenvproc.bat 56  
setenvproc.sh 56  
shared disk  
    setting up 142  
SHAREDYNAMIC 42, 84  
shipping

- log 114
- single point of failures 98
- single points of failure 92
- single sign-on 127
- SMS 171
- Software 126
- software availability 238
- space management 42
- SPOF 92, 98
- SQL2413N 24
- SQLLIB 21
- SRDF 224–225
- SSL 65, 76
- SSL key database 70
- stager 33, 36, 48, 68, 170, 186
- staging area 31, 35
- staging area properties 36
- STANDARD management class 74
- standby 105
- standby adapter 151
- standby node 126
- standby systems 38
- statement failure 9
- storage area 39–40, 42
  - Content Manager 30
- Storage Area Network 207
- storage areas 65
- storage class 39
- storage classes 37
- storage pool 88
- storage pools 57
- stored object 43
- su 138
- Sun Cluster 105
- SUSPENDSERVERTIME 49
- Symmetrix 224
- Symmetrix Remote Data Facility 224
- system volume 39

## T

- tablespaces 14
- tape
  - backup 25
- TCO 203
- terrorist attacks 9
- TEVENTIXnnnnnn 67
- text index frequency 52
- text search options 51

- time stamp 23
- Timefinder 238
- Tivoli Storage Manager 3, 21, 25, 27, 30, 32, 35, 40–41, 57, 70, 72, 87, 89, 108, 112, 168, 178–179, 187, 236
  - backup 79
  - restoring 81
  - space reclamation 42
- Tivoli Storage Manager API 40, 75
- Tivoli Storage Manager Backup Archive client 25
- Tivoli Storage Manager central scheduler 80
- Tivoli Storage Manager client 43–44
- Tivoli Storage Manager client options file 76
- Tivoli Storage Manager database backup 88
- Tivoli Storage Manager definitions 180
- Tivoli Storage Manager file system backup 44
- Tivoli Storage Manager scheduler 44, 88
- Tivoli Storage Manager server 21, 41, 43, 57, 89, 123
- Total Cost of Ownership 203
- TRACKMOD 26–27
- transaction 15
- transaction failure 9
- transaction reconciliation utility 38
- transactions 14, 19–20
- TRebuildCompTypeICM 53
- triangular architecture
  - Content Manager 2
- TSM\_MGMTCLASS 77
- two-way replication 112

## U

- unit of work 15
- unmount 143
- user error 8
- user exit 20
- user ID 129
  - creation 135
- USEREXIT 16, 24

## V

- validation discrepancy report 216
- validation utilities 55, 211
- varyoffvg 143
- Veritas 228, 236
- VG 126
  - import 145
- VideoCharger 3, 29, 32, 36–37

VLANS 239  
volhist.out 89  
volume group 126  
    create 130  
    import 145  
Volume Validation Utility 56  
VPN 241

## **W**

warm backup 13  
Web application server 107  
WebSphere 32, 56  
WebSphere Application Server 4, 30, 65, 98, 108,  
170  
    cloning 224  
WebSphere cloning 124  
WebSphere Edge Server Network Dispatcher 107  
WebSphere MQ 58, 220, 222, 224  
WebSphere MQ Workflow 58  
workflow 222  
workflow routing 2



## Content Manager Backup/Recovery and High Availability: Strategies, Options, and Procedures

(0.5" spine)  
0.475" <-> 0.875"  
250 <-> 459 pages







**Redbooks**

# Content Manager Backup/Recovery and High Availability: Strategies, Options, and Procedures

**Introducing basic  
concepts, strategies,  
options, and  
procedures**

**Addressing business  
continuity and  
disaster recovery  
issues**

**Providing practical  
case studies**

Structured and unstructured data is constantly growing, data retention requirements and user access requirements are continuously changing, and the demand for the readiness and availability of business systems and data becomes even higher. The use of content management systems is vital and necessary; it is what makes an organization's success viable. The *availability* of these systems is of crucial importance.

Several technologies of various degrees have provided an answer to backup, availability, and disaster recovery requirements, but all at a price. How can you achieve maximum availability of your IBM DB2 Content Manager systems while balancing costs, resources, and skills?

The purpose of this IBM Redbook is to introduce the concepts of backup/recovery, high availability, and disaster recovery for Content Manager systems, and provide strategies, options and implementation steps to protect your Content Manager systems. We also explore, through various case studies, how to apply your newly gained knowledge to real-world Content Manager system implementation and practices. This Redbook will also help IT architects, specialists, project managers, and decision makers identify the best high availability and disaster recovery strategies and integrate them into the Content Manager solution design process.

**INTERNATIONAL  
TECHNICAL  
SUPPORT  
ORGANIZATION**

**BUILDING TECHNICAL  
INFORMATION BASED ON  
PRACTICAL EXPERIENCE**

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

**For more information:**  
[ibm.com/redbooks](http://ibm.com/redbooks)