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

by Wei-Dong Jackie Zhu et al. ISBN:0738498203

IBM Redbooks © 2004 (274 pages)

The purpose of this 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.

Table of Contents

< Day Day Up >

Back Cover

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.

< Day Day Up >
< Day Day Up >

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

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

Redbooks
ibm.com/redbooks

Introducing basic concepts, strategies, options, and procedures

Addressing business continuity and disaster recovery issues

Providing practical case studies

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).

ISBN:0738498203

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

< Day Day Up >
< Day Day Up >

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.

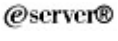
IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

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.

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®
	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.

< Day Day Up >
< Day Day Up >

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 Juschkewitz
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

Chapter 1: 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.

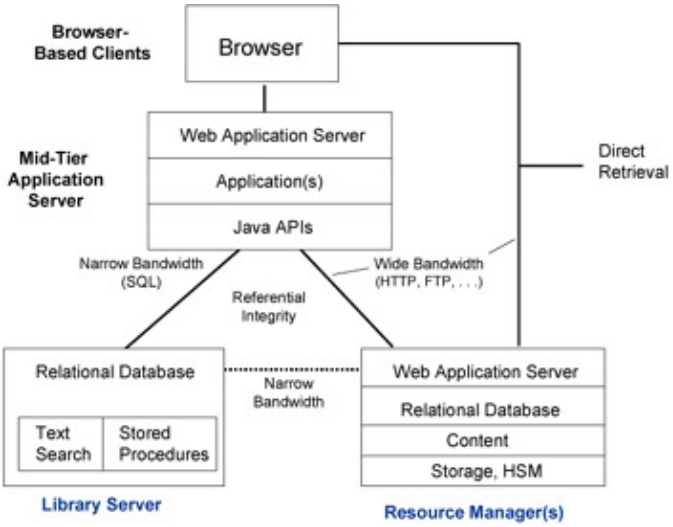


Figure 1-1: Content Manager Version 8 logical model

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.

< Day Day Up >

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.

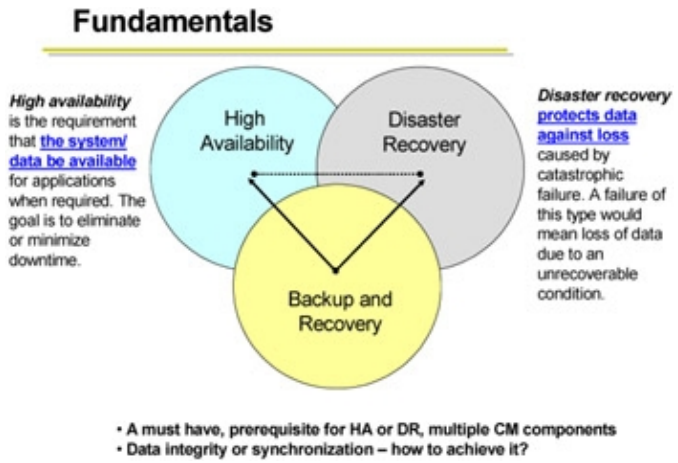


Figure 1-2: Fundamentals

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.

Chapter 2: Backup and Recovery Strategies and Options

Overview

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 detain 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.

< Day Day Up >
< Day Day Up >

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.

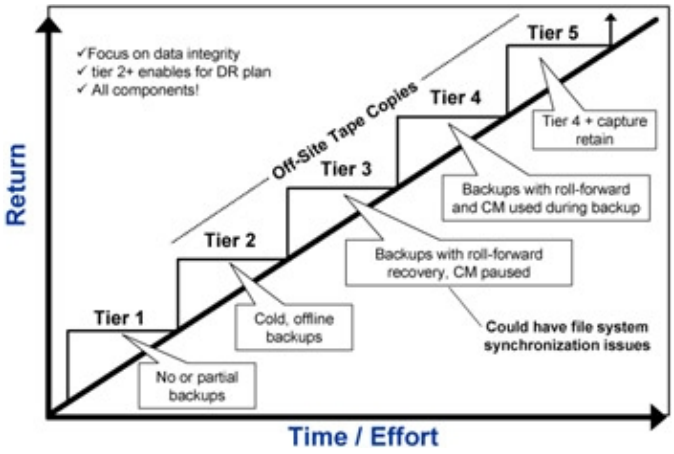


Figure 2-1: Backup tiers for Content Manager

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.

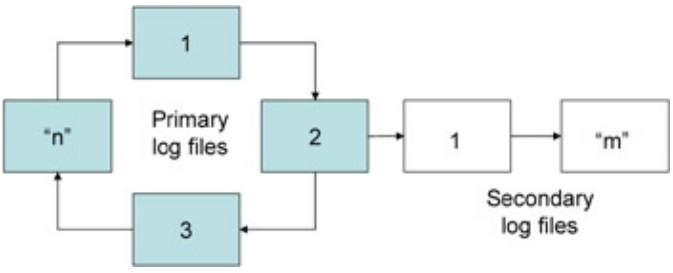


Figure 2-2: Circular logging

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) = MAC0

 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\SQLOGDIR\
 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 chckpt (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.

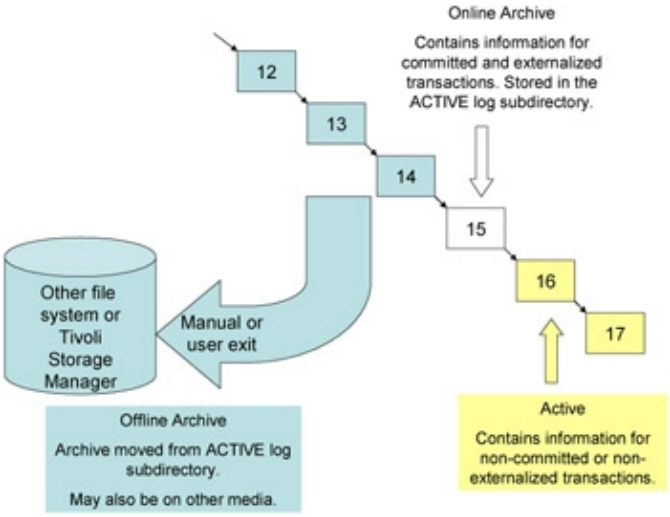


Figure 2-3: Archival logging

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:

   ```
   cl 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).