

IBM Content Manager for Multiplatforms



System Administration Guide

Version 8 Release 2

IBM Content Manager for Multiplatforms



System Administration Guide

Version 8 Release 2

Note

Before using this information and the product it supports, read the information in "Notices" on page 125.

Second Edition (March 2003)

This edition applies to Version 8 Release 2 of IBM Content Manager for Multiplatforms (product number 5724-B19) and to all subsequent releases and modifications until otherwise indicated in new editions.

Portions of this product are: Outside In[®] Viewer Technology © 1992--2000 Inso Corporation. All rights reserved.

© **Copyright International Business Machines Corporation 1993, 2003. All rights reserved.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About this guide	v	Step 7: Diagram data relationships.	45
Who should use this guide	v	Step 8: Decide whether you require a custom	
How to use this guide	v	data model	45
Where to find more information.	v	Step 9: Model your data in Content Manager	46
Information included in your product package	v		
Support available on the Web	vi		
How to send your comments	vii		
What's new in Version 8	vii		
Chapter 1. Getting started	1	Chapter 3. Defining and configuring	
Setting up First Steps to understand the system		servers	51
administration client.	1	Defining a library server	51
Administering the system	1	Connecting to a local and remote database	51
Display names.	3	Configuring a library server	53
Defining and configuring servers	3	Allowing trusted logon	54
Managing object storage	3	Adding a resource manager to the library server	54
Managing servers.	4	Changing the library server and system	
Managing user access	4	administrator's password to the resource	
Managing object retrieval	4	manager	54
Managing databases.	4	Changing the database access passwords	55
Managing document routing	4	Defining language codes	55
Administrative domains	5	Defining a resource manager	57
The system administration client.	5	Configuring a resource manager	58
Logging on to the system administration client	5	Configuring Secure Sockets Layer	58
Switching product views	6	Cataloging objects from your local machine	60
Switching library servers	6	Configuring a media server	61
Switching federated servers	6	Staging area	61
Keyboard access	7		
Chapter 2. Modeling your data in		Chapter 4. Managing object storage	63
Content Manager	9	Device managers	64
Building blocks for data models that are provided by		Storage classes	64
Content Manager.	9	Storage systems	65
Attributes	10	Storage groups	66
Components	12	Migration policies	66
Item types.	15	Collections	67
Items	21	Replication	67
Semantic types	21	Creating server definitions	67
Forming relationships between items	22	Library server monitor fail-over service	68
Objects	26	Turning on replication for objects that have	
Text search	29	already been stored.	68
Modeling sample data structures	32	Defining replication rules in administrative	
Scenario 1: Applying the building blocks	32	domains	70
Scenario 2: Modeling automobile insurance data	34	Lan cache	70
Modeling your data	37		
Step 1: Identify your data.	37	Chapter 5. Managing servers	73
Step 2: Separate your data into operational and		Starting and stopping servers	73
non-operational	39	Starting and stopping a Windows server.	73
Step 3: Sort your data into like types	40	Starting and stopping an AIX server	75
Step 4: Identify your users and what data they		Starting and stopping a server on the Solaris	
need to access	41	Operating Environment	76
Step 5: Within each data type, identify the		Synchronizing servers	77
elements that might be searched for	42	Backing up and restoring your data	77
Step 6: Identify hierarchies and elements that		Tracing errors.	78
might have multiple values	43	Replacing or repartitioning a hard disk	78
		Chapter 6. Managing resource manager	
		utilities and services.	81

General configuration of resource manager utilities and services	81
Configuration for AIX and Solaris	81
Configuration for Windows	81
Resource manager services	82
Configuring the resource manager services on AIX or Solaris	82
Starting and stopping resource services on AIX or Solaris	82
Asynchronous Recovery utility overview	83
Configuring the asynchronous recovery utility.	84
Asynchronous utility logging	84
Running the asynchronous recovery utilities on Windows	84
Running the asynchronous recovery utilities on AIX	84
Running the asynchronous recovery procedure on a Solaris Operating Environment system	85
Overview of validation utilities.	85
Configuring the validation utilities	85
Working with the resource manager/library server validation utility	86
The resource manager volume validation utility	88

Chapter 7. Managing user access 91

Creating user IDs and passwords	91
Understanding DB2 administration authority	92
Connecting to DB2 using the INI files	92
Changing the library server and system administrator's password to the resource manager	93
Changing the database access passwords	93
Importing users from LDAP.	93
Introducing privileges	94
Creating privilege sets.	95
Creating privilege groups.	95
Assigning a privilege set to a user.	95
Assigning a user ID a grant privilege set	96
Assigning users to resource managers	96
Assigning users to collections	96
Creating user groups	96
Creating access control lists	96
Assigning a privilege set to an access control list	97
Creating domains	97
Administering domains	98
Accessing domains	98
Assigning a user to a domain	98
Assigning a user group to a domain	99
Assigning a privilege set to a domain	99
Assigning a resource manager to a domain.	99
Assigning a collection to a domain	99
Moving a user from one domain to another	99

Moving a user group from one domain to another	100
Moving a resource manager from one domain to another	100
Moving a collection from one domain to another	100
Moving a privilege set from one domain to another	101
Moving an access control list from one domain to another	101

Chapter 8. Managing databases 103

Optimizing server databases	103
Optimizing a DB2 database.	103
Removing entries from the events table.	104
Migrating objects	105
Creating a migration policy.	105
Setting up remote migration	106
Changing the date of migration	106
Migrating and purging the VideoCharger Server media objects at regular intervals.	107

Chapter 9. Managing document routing 109

Defining a process.	109
Defining work baskets	110
Defining collection points	110
Adding a work basket or collection point to a process	111
Branching in a process	111
Ad hoc routing processes	111
Defining worklists.	112
Defining work packages.	112
Creating folders for a process	112
Updating a process	112
Deleting a process	113

| ICM library server event table log. 115

Accessibility features 123

Keyboard input and navigation	123
Features for accessible display.	123
Compatibility with assistive technologies	124
Accessible documentation	124

Notices 125

Trademarks	127
----------------------	-----

Glossary 129

Index 143

About this guide

This guide provides an overview of how to administer your Content Manager Version 8 Release 2 system. It:

- Provides an overview of administration tasks and describes the tools that are available to assist you in the performance of those tasks.
- Identifies the information that you need to get your Content Manager system up and running.
- Summarizes the required tasks for maintaining your system.

Who should use this guide

Use this guide if you are the system administrator who is responsible for setting up and maintaining the Content Manager system for your enterprise. This guide provides conceptual information for understanding those tasks. For information about completing specific tasks, see the Content Manager system administration client online help.

How to use this guide

This guide assumes that you are using the system administration client that came with the Content Manager Version 8 Release 2 product. If you want to create a system administration client to suit your enterprise, or add function to the system administration client using the APIs, see the *Workstation Application Programming Guide* or the online Application Programming Reference.

The general term "Windows[®]" applies to Microsoft[®] Windows NT[®] 4.0 and Windows 2000.

For specific information about how to use the system administration client, see the online help. The online help provides detailed information about the fields and functions that are associated with each window.

Where to find more information

Your product package includes a complete set of information to help you plan for, install, administer, and use your system. Product documentation and support are also available on the Web.

Information included in your product package

The product package contains an information center and each publication in portable document format (.PDF).

The information center

The product package contains an information center that you can install when you install the product. For information about installing the information center see *Planning and Installing Your Content Management System*.

The information center includes the documentation for Content Manager, Enterprise Information Portal, and VideoCharger. Topic-based information is

organized by product and by task (for example, Administration). In addition to the provided navigation mechanism and indexes, a search facility also aids retrievability.

PDF publications

You can view the PDF files online using the Adobe Acrobat Reader for your operating system. If you do not have the Acrobat Reader installed, you can download it from the Adobe Web site at www.adobe.com.

Table 1 shows the Content Manager publications included with IBM Content Manager for Multiplatforms.

Table 1. Content Manager publications

File name	Title	Publication number
install	<i>Planning and Installing Your Content Management System¹</i>	GC27-1332-01
migrate	<i>Migrating to Content Manager Version 8</i>	SC27-1343-01
sysadmin	<i>System Administration Guide</i>	SC27-1335-01

When you order IBM Content Manager for Multiplatforms, you also receive IBM Enterprise Information Portal for Multiplatforms. Or, you can separately order IBM Enterprise Information Portal for Multiplatforms. Table 2 shows the Enterprise Information Portal publications that are included with the product.

Table 2. Enterprise Information Portal publications

File name	Title	Publication number
apgwork	<i>Workstation Application Programming Guide¹</i>	SC27-1347-01
ecliinst	<i>Installing, Configuring, and Managing the eClient</i>	SC27-1350-02
eipinst	<i>Planning and Installing Information Integrator for Content</i>	GC27-1345-01
eipmanag	<i>Managing Information Integrator for Content</i>	SC27-1346-01
messcode	<i>Messages and Codes²</i>	SC27-1349-01

Notes:

1. The *Workstation Application Programming Guide* contains information about programming applications for both Content Manager and Enterprise Information Portal.
2. *Messages and Codes* contains the messages and codes for Content Manager and Enterprise Information Portal.

Support available on the Web

Product support is available on the Web. Click **Support** from the product Web sites at:

www.ibm.com/software/data/cm/

www.ibm.com/software/data/eip/

The documentation is included in softcopy with the product. To access product documentation on the Web, click **Library** on the product Web site.

An HTML-based documentation interface, called Enterprise Documentation Online (EDO), is also available from the Web. It currently contains the API reference information. Go to the Enterprise Information Portal Library Web page for information about accessing EDO.

How to send your comments

Your feedback helps IBM to provide quality information. Please send any comments that you have about this publication or other Content Manager or Enterprise Information Portal documentation. You can use either of the following methods to provide comments:

- Send your comments from the Web. Visit the IBM Data Management Online Reader's Comment Form (RCF) page at:
www.ibm.com/software/data/rcf
You can use the page to enter and send comments.
- Send your comments by e-mail to comments@vnet.ibm.com. Be sure to include the name of the product, the version number of the product, and the name and part number of the book (if applicable). If you are commenting on specific text, include the location of the text (for example, a chapter and section title, a table number, a page number, or a help topic title).

What's new in Version 8

Version 8.2: Version 8.2 includes a variety of enhancements from Version 8.1. Version 8.2 adds more workflow features to the eClient, increases resource management function, and supports the latest in database and client technology, including DB2 Universal Database Version 8.1, Oracle Version 8.1.7.4 and Version 9.2.0.1, and WebSphere Version 5. These highlights, and other enhancements to the Version 8.2 product, are summarized below:

Enterprise Information Portal name change to IBM Information Integrator for Content

Enterprise Information Portal has been renamed to Information Integrator for Content. Although the names of the books have changed for Version 8.2, the text within the books continues to show the product name Enterprise Information Portal. When searching the Web for more information, you can continue to use Enterprise Information Portal, or EIP, until the transition to the new name is complete.

Support for Oracle Version 8.1.7.4 or Version 9.2.0.1 or later

Content Manager V8.2 adds support for Oracle databases managing the metadata stored in both library server and resource manager. Migration tools are included for Oracle users of Content Manager Version 7. **Note:** Oracle does not manage Enterprise Information Portal database server contents.

Replication

Content Manager V8.2 includes resource manager replication, which is the ability to store objects in multiple locations, managed by replication resource managers. Object replicas will behave as LAN cache objects for improved load balancing.

LAN cache

LAN cache support in Content Manager V8.2 provides application-transparent caching, using local servers as defined by the system administrator.

Support for DB2 UDB V8.1

Content Manager V8.2 and Enterprise Information Portal V8.2 supports DB2/UDB V8.1. The connection concentration feature of DB2 V8.1 provides increased scalability for two-tier applications and clients (such as the Content Manager V8 Client for Windows). DB2/UDB V8.1 has replaced the DB2 Universal Database Text Information Extender (TIE) with Net Search Extender (NSE).

Support for WebSphere Application Server Version 4 and Version 5

WebSphere Application Server Version 5 introduces server deployment and data access and management from any web browser.

Federated folders

eClient now has the ability to organize documents and native folders from multiple repositories into a single federated folder and start that folder on a workflow. Federated folders also allows users to persistently store search results in the EIP federated database where users can retrieve them at any time. Full CRUD (create, retrieve, update, and delete) operations are available against these federated folders without re-indexing.

Advanced workflow collection points

Workflow is now fully supported on AIX and Solaris. The workflow builder, APIs, Collection Points Monitor, and JavaBeans provide improved workflow function and usability.

Microsoft Visual Studio .NET for building applications

The Content Manager and Enterprise Information Portal 8.1 and later APIs now support Microsoft Visual Studio .NET for writing content management applications or to integrate applications built using Microsoft Visual Studio .NET.

Version 8.1: Version 8.1 begins a legacy of integration and versatility. One of the many highlights and improvements from previous Content Manager products is the new data model structure which allows for more document customization. The changes to the Content Manager product in Version 8.1 are summarized below:

Improved performance

The library server and resource manager use DB2 stored procedures and leverage DB2 technology to significantly reduce network traffic and improve performance and scalability.

Support for Sun Solaris

Both the library server and resource manager can be installed on Sun Solaris.

Enhanced data model

The new hierarchical data model provides the basis for customized compound document management solutions.

Improved workflow

Through integrated document routing, workflow capabilities have been improved with sequential routing, dynamic routing, and collection points.

Integrated text search

In addition to attribute-based searching, client users can now perform full-text searching on text-based document information. The text search

function now uses the DB2 Universal Database Text Information Extender, which contributes to a streamlined process for setting up text searching.

Common system administration

A single client application provides separate access to Content Manager and Enterprise Information Portal. Within Content Manager, administrative domains provide a way to limit administrative access to subsections of the library server.

Full-function desktop client and enhanced eClient

Client enhancements provide users with an out-of-the-box application for rapid deployment or line of business application integration. The Client for Windows supports integrated text search, document routing, the hierarchical data model (to a single child component level), versioning, and index during import. The eClient includes integrated text search, EIP advanced workflow, version control, and multi-valued attributes.

Easier installation

Installation is consistent across supported operating systems and customized installation information is provided by the Start Here CD's Planning Assistant. Silent and console installations are also provided.

Information center

The browser-based information center includes the documentation for Content Manager, Enterprise Information Portal, and VideoCharger. Topic-based information is organized by product and by task (for example, Administration). In addition to the provided navigation mechanism and indexes, a search facility also aids retrievability.

Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features for this product include:

- The ability to operate all features using the keyboard instead of the mouse.
- Support for enhanced display properties.
- Options for video and audio alert cues.
- Compatibility with assistive technologies
- Compatibility with operating system accessibility features
- Accessible documentation formats

PeopleSoft and Siebel integrations

Users of PeopleSoft and Siebel applications can now configure these applications to access content stored in a variety of content servers using the eClient.

Chapter 1. Getting started

The system administration client provides the tools that you need to set up and manage your system. Use this information to understand these tools and the tasks that you must complete at a high level. Use the help provided for each window for field-level details and step-by-step instructions for completing each window. For more system administration documents, click **Support** from the product web sites: www.ibm.com/software/data/eip and www.ibm.com/software/data/cm.

Setting up First Steps to understand the system administration client

First Steps is a module that comes with every installation of Content Manager. *First Steps* provides you with sample data and populates objects so that you do not have to use real data. Use *First Steps* if you want to explore item types, user access, and document routing to help you understand the basic structure, look, and feel.

You can get to First Steps through the shortcut menu bar, **Start → Programs → IBM Content Manager for Multiplatforms → First Steps**. Read the First Steps information and load the sample data. Then, you can begin working with the sample data by creating item types, adding users and user groups, and examining the data model structure.

Administering the system

As the system administrator, you must complete one or more of the following tasks:

- Defining and configuring servers
- Managing servers
- Managing object storage
- Managing user access
- Managing object retrieval
- Managing databases
- Managing document routing

Figure 1 on page 2 summarizes the operations that are involved in defining, configuring, and managing your library server and resource manager:

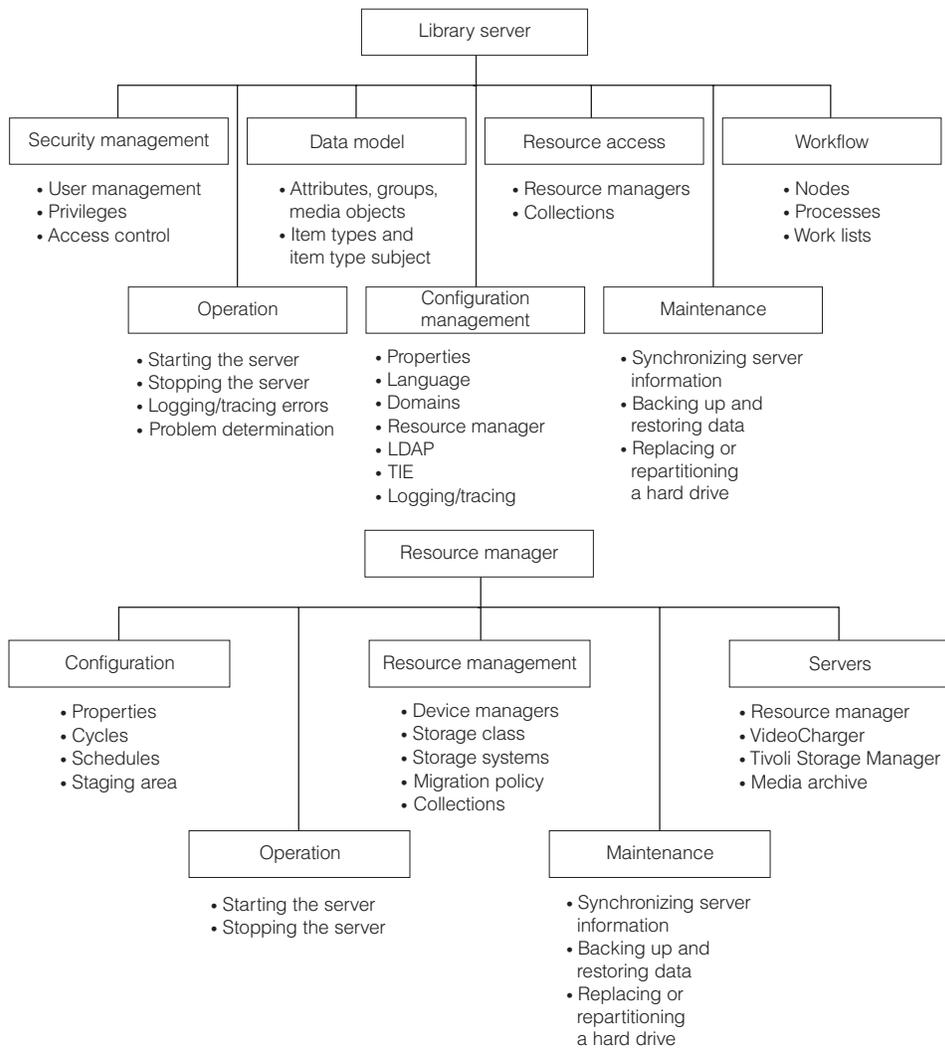


Figure 1. High-level view of operations used to define, configure, and manage both the library server and the resource manager

You can find more information about:

- Security management, which includes privilege sets and administrative domains, in Chapter 7, “Managing user access”, on page 91.
- The data model, which includes reference attributes and links, in Chapter 2, “Modeling your data in Content Manager”, on page 9.
- Resource access and resource management, which include SMS concepts and database management, in Chapter 4, “Managing object storage”, on page 63 and Chapter 8, “Managing databases”, on page 103.
- Resource manager and library server configuration, in Chapter 3, “Defining and configuring servers”, on page 51.
- Servers, which include storage management and file systems, in Chapter 4, “Managing object storage”, on page 63.
- Workflow, which includes document routing, in Chapter 9, “Managing document routing”, on page 109.
- Operation and maintenance, which include reconciling discrepancies between the library server and resource manager, in Chapter 5, “Managing servers”, on page 73.

For help with completing the windows in the system administration client, use the online help.

Important: In the user interface, an asterisk (*) indicates a required field. If you attempt to save an object that contains an empty required field, Content Manager displays an error.

Display names

Certain windows, like the ones for attributes, item types, and MIME types, require a name and a display name. The **Name** field identifies an object to the system administration client while the **Display name** field identifies the name that end users see when they work with the end user application.

Important: keep display names distinct from other display names. If you do not, you can confuse the end-user. For example, you can have an attribute for someone's first name and use a display name of Name. You could also put Name as the display name of a person's last name. End users will see two attributes called Name, but they will not know which value to enter for each attribute.

Defining and configuring servers

At this point, you must have already installed or migrated your system to Content Manager Version 8 Release 2. See *Planning and Installing Your Content Management System* if you have not completed these tasks. Defining servers and defining the relationships between them establishes the basis for your system.

You need to decide how to set up your library server, like:

- Whether or not to use LDAP or Text Information Extender (TIE)
- What type of access users have to the system (default ACLs) and its objects (item level, item type level, mixed, or library level)
- What detail you want to trace errors
- Whether or not to log system administrator events

You also need to decide how to set up your resource manager, like:

- When and how to migrate objects
- When to purge objects
- What servers a resource manager recognizes

Make sure that you have configured your library server to recognize and connect to the databases that you need to access. For information about defining and configuring servers, see Chapter 3, "Defining and configuring servers", on page 51.

Managing object storage

You need to define enough space to store the content that your users want to store. You need at least one resource manager defined and connected to the library server to begin defining space for object storage. See Chapter 3, "Defining and configuring servers", on page 51 for more information about defining a resource manager.

To define object storage, you need to know how you want to group your objects for optimal performance. These groups are called collections. For each collection that you create, you need to define a storage group, a migration policy, a device manager, a storage class, and a storage system. See Chapter 4, "Managing object storage", on page 63 for more information.

Managing servers

Content Manager provides several ways to save, retrieve, and assure that information is available to you when you need it. When you use the utilities provided by Content Manager, you can recover information that you might have lost or back up information that you currently have.

For further information about how to save, retrieve, back up and recover information in the resource manager and the library server, see Chapter 5, “Managing servers”, on page 73.

Managing user access

You allow users access to the Content Manager system by creating user IDs and privileges. Each user needs a user ID and password. You restrict access to the data stored in the system by defining and assigning appropriate privileges to the users.

You can read more about managing user access, including the concepts of administrative domains and LDAP, in Chapter 7, “Managing user access”, on page 91.

Managing object retrieval

Object retrieval begins with creating item types that provide order to all of the information needed to efficiently run businesses and keep customers satisfied. When you give structure to information, the information becomes more useful and easier to retrieve. With Content Manager, you can create simple item type structures, or you can create item type hierarchies, to define more complex and detailed relationships. For more information about item types, see Chapter 2, “Modeling your data in Content Manager”, on page 9.

Managing databases

You continue to monitor database performance throughout the life of the Content Manager system. You work in conjunction with database administrators to ensure that the system runs with optimum performance. To manage the Content Manager database, see Chapter 8, “Managing databases”, on page 103.

Managing document routing

Content Manager not only provides a robust system to store information, it also provides a system to move stored information into the hands of those people who need to use the information. Document routing is a powerful and convenient tool that you can decide to use.

Document routing is a work management tool that you use to direct documents from one user to another. Based on their privileges, users inspect documents and update them to complete a work step. For example, XYZ Insurance uses document routing for their auto claim process. In the process, work is directed from an insurance clerk to an underwriter. An underwriter waits for the police report and the insurance adjuster’s damage assessment and then directs the claim to an insurance accountant or an underwriter assistant, depending on whether the underwriter approves or rejects the claim. Document routing allows XYZ Insurance to approve a claim without using paper or manually carrying a claimant’s folder from one person to another.

If you plan on using document routing, you must enable it. For more information about document routing and enabling the tool, see Chapter 9, “Managing document routing”, on page 109.

Administrative domains

Content Manager allows you to create divisions, or domains, of the library server exclusive to a group of users. Each domain has one or more administrators that manage user access within that domain.

You do not need to have domains to have a secure system. You might, however, consider using administrative domains if you have a large user base divided among many departments or you manage the library server for more than one company. For example, XYZ Insurance might want to divide the company by department because users in the Claims department do not need to view or work with any documents in the Sales department. After you enable administrative domains, you cannot disable them and you need to restart the system administration client to see the effect of enabling them.

For more information about creating domains, see “Creating domains” on page 97.

The system administration client

You can use the system administration client for most of your administration tasks. With the system administration client, you can access both the Content Manager and Enterprise Information Portal products from one user interface, if you have installed both products.

You can switch products without logging off from one and logging on to another. You can also switch library servers without logging off and on again.

After you log on to the system administration client, you can use keyboard access keys to navigate within it.

For additional information about using the system administration client, see the online help that is available from every window.

Logging on to the system administration client

From the system administration client, you can access both the Content Manager and Enterprise Information Portal system administration databases without having to log off and on again. You can also start the system administration client from more than one location by using the same user ID. You can start multiple clients from the same machine or different machines. **Requirement:** You must be using a library server configuration that allows multiple logon.

To start the system administration client, complete the following steps:

1. From the Windows taskbar, click **Start** → **Programs** → **IBM Content Manager for Multiplatforms®** → **System Administration**. You can resize the logon window if necessary after you open it.
2. Select Content Manager or Enterprise Information Portal as the server type to log in to first.
3. Select a library server.
4. Type a valid user ID and password. A user ID has one to 32 alphanumeric characters and is not case-sensitive. A password has one to 16 alphanumeric

characters and is case-sensitive. You might not have to enter a user ID and password if you have set up your logon process to use the workstation user ID and password.

5. Click **OK**.

Important: When you attempt to start the system administration client, the system checks to see if the single-sign on option was selected during installation. If it was selected during installation, single-sign on is active.

- If the single-sign on option is not active, type the user ID and password when logging on. The default user ID is icmadmin.
- If single-sign on is active, the program does not ask for the user ID or password, but looks at the user ID that is logged on to the system at that time. This is the user ID that was defined during the installation of the library server, for example, icmadmin, the default user ID. If you are not logged on to the system with that user ID, the system administration client cannot open. The workaround is to log on to the system with the same user ID that was defined at the beginning of the installation program for the library server.

Stopping the client: To stop the system administration client, close the System Administration window.

Switching product views

Previously, if you had Content Manager and Enterprise Information Portal as part of your enterprise solution, you had to open two separate system administration clients. In the current version, if you have both products installed, you can administer both systems from the same user interface. Switching from one system administration view to the other provides a convenient way to modify information that applies to both systems and fast access to either product.

To switch from one product to the other without logging off, go to the main system administration window and use the pull-down menu above the left pane. If the pull-down menu lists any product other than the one that you are currently using, you can switch to that product.

Switching library servers

If you plan to switch library servers while you are logged on to the system administration client, you must ensure that the database that your library server uses is running. The database is an integral part of the library server, so the library server comes up and goes down with its associated database.

To switch library servers, select the library server that you want to work on in the left pane. Notice that the list of resource managers has changed accordingly (expand **Resource Managers** in the left pane of the system administration client main window). Resource managers can associate to only one library server, so each time you change library servers, you get access to a new list of resource managers.

Switching federated servers

If you have Enterprise Information Portal as part of your business solution, you can switch from Content Manager administration to Enterprise Information Portal administration. Within the Enterprise Information Portal administration, you can also switch servers.

For more information about federated servers and Enterprise Information Portal, see *Managing Information Integrator for Content*.

Keyboard access

You can use the keyboard to access all of the functions of the system administration client. In general, access from the keyboard follows standard Microsoft guidelines. For example, you can open the **File** menu from the keyboard by holding down the Alt key and pressing F. Access from the keyboard differs from standard Microsoft guidelines in the following ways:

Access keys, tabbing, and tables

Access keys are provided only for buttons and menu items. Press Tab to reach fields that do not have a shortcut key combination.

Within a table, the Tab key moves the cursor to the next cell. To move out of the table to the next field, hold down the Ctrl key and press Tab. When the cursor is within a table, pressing enter is not equivalent to clicking **OK**; you must move out of the table first.

Menus

Pressing Alt+Spacebar does not open the **Program** menu from the left icon on the title bar of the Content Management System Administration window. Pressing Shift+F10 does not open pop-up menus. You can access pop-up menu functions from the **Selected** menu.

Tree views

You can expand or collapse a tree by pressing Enter or by using the left and right arrow keys. Pressing the * key does not expand a tree selection. Pressing the plus and minus keys on the numeric key pad does not expand or collapse the tree. Typing characters or pressing Backspace while on the tree does not select an item.

List boxes, check boxes, and radio buttons

In a list box, press the Down Arrow and Up Arrow keys to select an item.

To select multiple sequential items

Hold down the Shift key while pressing the Down or Up Arrow key.

You cannot select items within a list box, list view, or tree by typing the characters of its name.

Within list boxes, the following actions have no effect:

- Pressing the Ctrl key with Page Up, Page Down, Home, or End
- Pressing a letter key
- Pressing Shift+F8

You can select individual radio buttons by pressing the Tab key and then the Spacebar, or by using the access keys. Arrow keys do not select radio buttons within a group.

Notebook tabs

Access keys are not provided for notebook tabs. Move the focus to a page tab using the Right and Left Arrow keys or the Tab key, or by pressing Ctrl+Page Down or Ctrl+Page Up.

Additional keystrokes

The following keys have no effect on text fields:

- Alt+Backspace
- Ctrl+Z
- Shift+Delete

Chapter 2. Modeling your data in Content Manager

This section discusses how to model your data using IBM Content Manager. Specifically, it describes:

- Basic conceptual building blocks for data modeling that are provided by Content Manager
- Scenarios that illustrate how to model sample data in Content Manager and the best methods for implementing your data model in different situations
- Step-by-step instructions for modeling your data in Content Manager

More information about data modeling concepts is also provided in the ICM API Education Samples. If you install Enterprise Information Portal, see the Getting Started section of the ICM Samples readme (README_SAMPLES_JAVA_ICM.txt or README_SAMPLES_CPP_ICM.txt), located in the X:\CMBROOT\Samples\java\icm or X:\CMBROOT\Samples\cpp\icm directory.

Building blocks for data models that are provided by Content Manager

This section describes the following data model building blocks that Content Manager provides:

- Attributes
- Components
- Item types
- Items
- Ways to form relationships between items
- Objects

Most of these building blocks include additional elements that are described in the appropriate subsections. To model your data, you must first understand these building blocks.

Restrictions: Certain data model elements that are described in this section might not be supported in the provided clients: the Client for Windows or the eClient. Table 3 lists the data model elements that are described in this section and whether they are supported in the clients.

Table 3. Client support for data model elements

Data model element	Supported by:	
	Client for Windows	eClient
Attribute	Yes ¹	Yes ¹
Attribute group	No	Yes
Root component	Yes	Yes
Child component	One level only	One level only
Item type classification: item	No	No
Item type classification: resource item	No	No
Item type classification: document	Yes	Yes

Table 3. Client support for data model elements (continued)

Data model element	Supported by:	
	Client for Windows	eClient
Item type classification: document part	Yes ²	Yes ²
Versions	Yes	Yes
Media object class	Yes	Yes
Item type subset ³	Yes	Yes
Semantic type	Yes ⁴	Yes ⁴
MIME type	Yes	Yes
Links	Folder only	Folder only
References	No	Can be displayed
Foreign keys	No	No

Notes:

1. Except for BLOB and CLOB types.
2. Client users are unaware of the presence of document parts. Creating document parts using user-defined document part types is not supported.
3. Referred to as “views” in the Client for Windows.
4. Semantic type support in the provided clients is transparent to the user. The clients do not provide a way for users to select from available semantic types.

Attributes

An *attribute* stores units of data (metadata) or values that describe a certain characteristic or property (for example, first name, surname, age, city, and so forth) of an item. The attribute can be used to locate that item. In earlier releases of Content Manager, attributes were called key fields.

You can create attributes from the main window of the system administration client or from the Attributes page in the Item Type Definition window. To create an attribute, you have to analyze the expected values for that attribute. For example, if you expect the value of an attribute to contain alphanumeric characters, then you could assign the attribute a variable character attribute type. Furthermore, you need to decide the maximum and minimum length for the variable character attribute value.

Restriction: If you specify that the attribute can contain a character large object (CLOB) or a binary large object (BLOB), consider that the Content Manager library server can only support up to 5 MB for the CLOB and BLOB attribute. The total amount of character or binary data that can be passed to the library server for creating or updating an item is 5 MB. Each character attribute requires 2 additional bytes in the buffer, and the buffer used for binary data also contains control information. In practice, the total amount of application data should be limited to less than 5 MB for each of these attributes. In developing an application using large attributes, you should consider whether these attributes should be implemented using objects on the resource manager.

You enable text search in the Library Server Configuration window. Then, if you want an attribute to be text searchable, you must select the **Text searchable** check box and specify the text search parameters. For example, you might decide that the attribute for customers’ first and last name in the Policy item type must be text

searchable so that a customer representative can use customers' first and last name to find their policies. However, you might find that making the Street attribute text searchable in the Policy item type is not important because street names are rarely unique and would not help the customer representative locate a specific policy.

Using the system administration client, system administrators define attributes in the window shown in Figure 2.

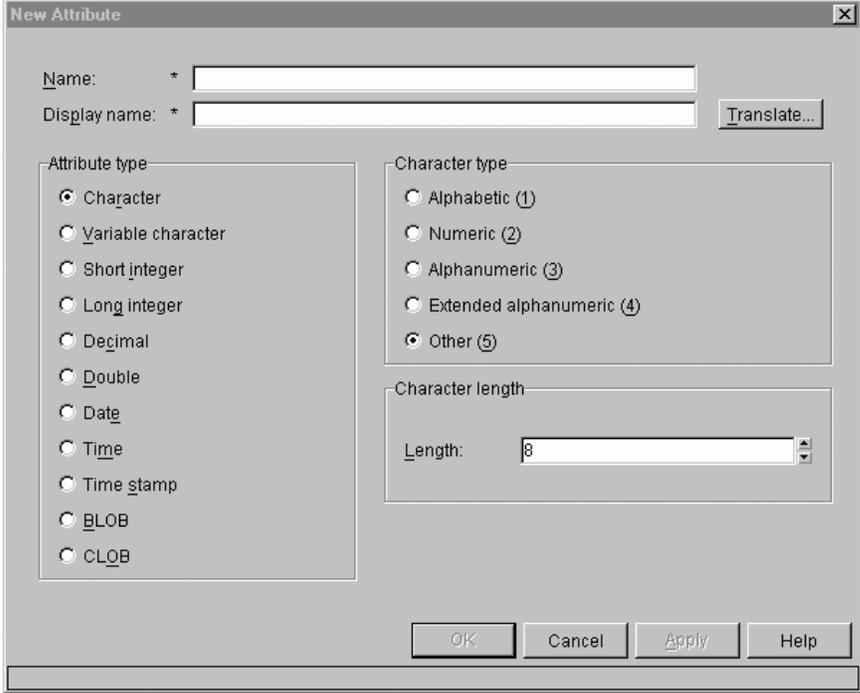


Figure 2. New Attribute window

The system administration client stores these defined attributes and makes them available for selection when you create or modify item types.

When creating attributes, you usually make them as basic as possible so that they are flexible enough to use throughout your system. You might find that you often use some of the same attributes together. For these attributes, you can create an attribute group. An *attribute group* is a set of attributes that are grouped together for convenience.

When you add an attribute group to an item type, all attributes in the attribute group are inserted into the item type at one time. For example, instead of inserting four attributes for every item type to create an address (street, city, state, and postal code), you can create an attribute group called Address that includes those four attributes. When you create an item type, you select the attribute group Address to get the attributes: Street, City, State, and PostalCode, as shown in Figure 3 on page 12.

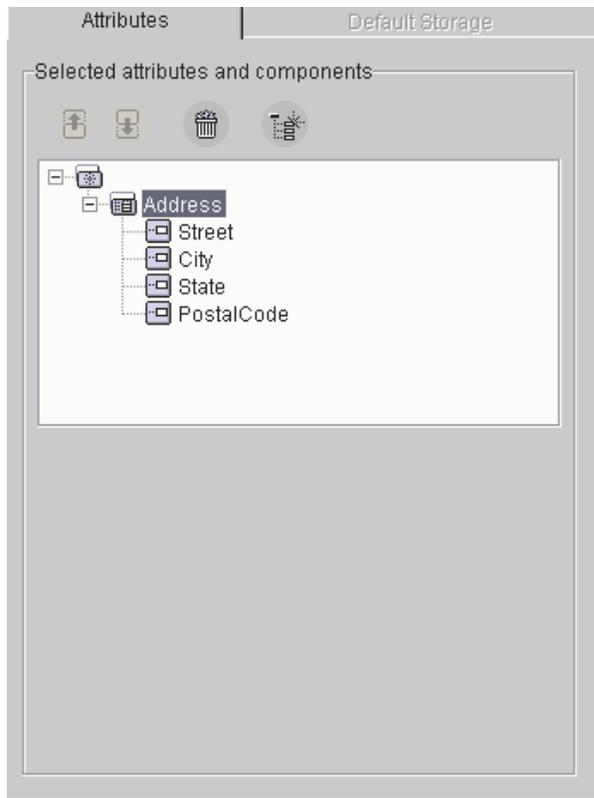


Figure 3. The Address attribute group in an item type definition

Note the blank area in the window below the list box that displays the attribute group. The area is blank because you cannot set any specific properties for the attribute group; the attribute group is only for the convenience of adding several attributes at once. To set properties for the attributes, you must select them individually.

Components

A *component* is a meaningful set of system-defined and user-defined attributes that you use to describe a type of data or some subset of it. There are two types of components, root and child. You can build item types by using one root component and zero or more child components.

In the underlying relational database, each component is represented by a table. Database indexing is available, and you define indexes at the component level.

The following sections describe the root and child components.

Root component

A *root component* is the first or only level of a hierarchical item type and consists of both system- and user-defined attributes. For example, a Personal auto policy item type might have a root component that includes the following user-defined attributes.

Policy number	Named insured	Named insured address	Vehicle make	Vehicle model	Vehicle Identification number (VIN)	...
---------------	---------------	-----------------------	--------------	---------------	-------------------------------------	-----

The hierarchical item type did not exist before Content Manager Version 8, so index classes that were created with earlier Content Manager versions were a single level with multi-valued attributes and index class subsets. In Content Manager Version 8, you can create a similar item type by creating one that has only a root component. Multi-valued attributes in Content Manager Version 8 are implemented as child components (see “Child component”). (Index class subsets are implemented as item type subsets see “Item type subset” on page 20.)

If you plan to use the hierarchical item type, you might change the root component somewhat to account for the child components that you plan to create. The example above might work well for a root component with no children; however, if you planned to create children, you might create this root component:

Policy number	Named insured	Named insured address	Insured vehicles	Operators	...
---------------	---------------	-----------------------	------------------	-----------	-----

Because a customer might insure more than one vehicle, the vehicle information, like the make, model, and Vehicle identification number (VIN), might be contained in a child component. Similarly, you might create a child component to store the multiple operators (residents in a customer’s home who can drive or operate the insured vehicles) that are insured under the policy.

Child component

A *child component* is the optional second or lower level of the hierarchical item type. Each child component is directly associated with the level above it. Use child components for detailed information for which multiple values might exist, information that previously (in earlier Content Manager releases) might have been contained in multi-valued attributes.

For example, Figure 4 on page 14 shows the Personal auto policy item type with two child components. One child component is for the vehicles that are insured under the policy. The other identifies the operators of the insured vehicles who are explicitly covered under the policy, for example, other members of the same household who can drive.

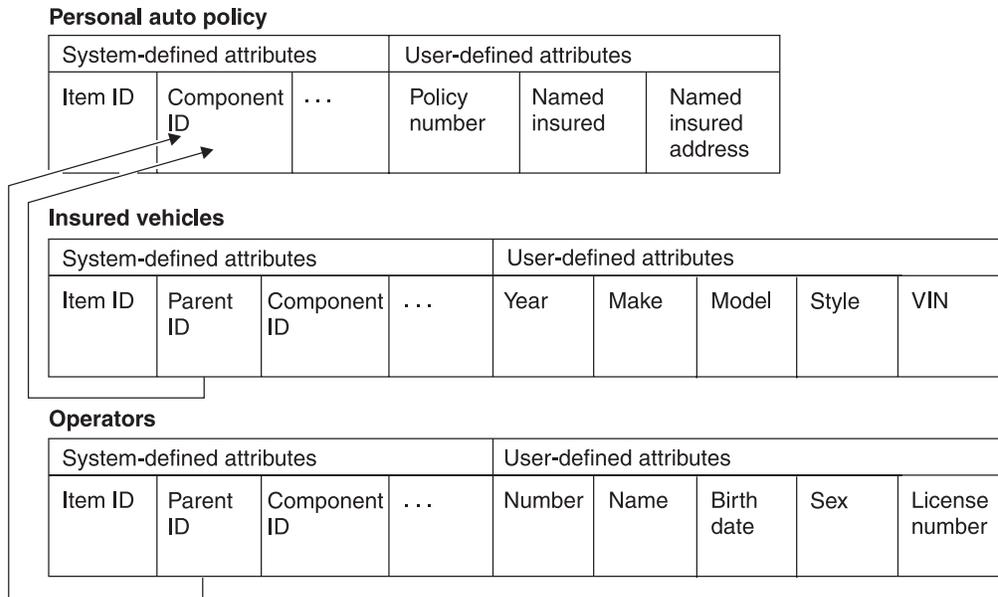


Figure 4. Item type with two child components. Parent IDs in the child components connect to the component ID in the root (or parent) component.

There is no limit to the number of component levels that you can create nor to the number of children that you can include at each level. However, if you plan to use the provided Client for Windows or eClient, be aware that these clients display only one child component level.

You create child components by clicking the child component icon (the fourth icon in Figure 5) on the Attributes page of the New Item Type Definition notebook.



Figure 5. Icon buttons on the Attributes page of the New Item Type Definition notebook. From left to right, the buttons perform the following actions: Move up, Move down, Remove, and Create child component.

After you click the child component button, the Attribute page changes so that you can set properties for the child component, as shown in Figure 6 on page 15.

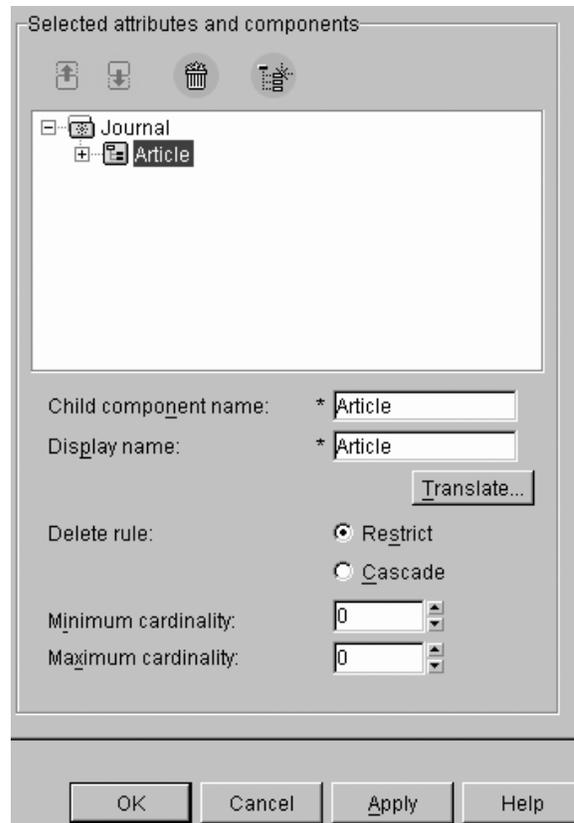


Figure 6. Defining a child component. You define a child component on the Attributes page in the New Item Type Definition notebook.

When you select a child component under **Selected attributes and components**, the fields are available for you to specify the following information:

- Name of the child component (**Child component name**).
- Name of the child component to display to client users (**Display name**).
- Whether to delete children of this child component (**Cascade**) if this child component is deleted. Note that this applies to created items that include this child component, not to the child component definition.

If you do not want to automatically delete children, click **Restrict**.

- The minimum and maximum number of rows in the database table that is created for this child component (**Minimum cardinality** and **Maximum cardinality**). For example, for the Operators and Insured vehicles child components, the minimum cardinality would be one because you cannot have an auto policy with no insured vehicles or drivers.

Although you specify a maximum cardinality, the storage space is not allocated until it is needed to store values.

Item types

An *item type* is a template for defining and later locating like items, consisting of a root component, zero or more child components, and a classification. The classifications are: item, resource item, document, and document part.

The template that you use to create specific items is the item type. By using the same template, items of the same type are consistently constructed, which helps

you to locate them and quickly define new ones. In Content Manager, you build item types for recording a consistent set of information about related items that you want to catalog.

For example, you might have an item type called Personal auto policy. The Personal auto policy item type includes a consistent set of characteristics, or attributes, for example: Policy number, Named insured, Named insured address, Vehicle make, VIN, and so forth. When you create an item of type Personal auto policy, you enter values for each of these attributes, and those values uniquely define that item.

The following sections describe item type classifications, media object classes, and item type subsets, all of which you must define when you define an item type.

Item type classifications

You must select one of four item type classifications when you create an item type, as shown in Figure 7.

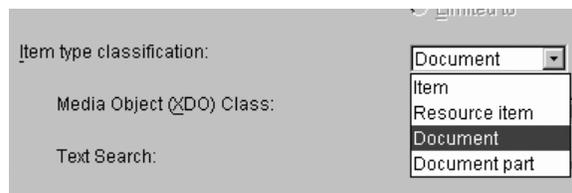


Figure 7. Selecting an item type classification. You select an item type classification on the Definition Page in the New Item Type Definition notebook.

By classifying the item type, you make a judgement about the purpose of the items created with this item type. The following sections describe the four item types classifications--item, resource item, document, and document part--in more detail.

Item type classification: Item: You use item types to create items. Although some items (resource items) can describe content that is stored on the resource manager, others are self-contained. Items are typically those things that you can describe completely with a set of attributes and are not a document or a file. Items are similar to a row in a database.

By classifying an item type as *item*, you determine that items of this type are self-contained, that they do not themselves describe separate stored content such as scanned documents, video, or audio. Examples of item types that you might classify as item:

- Customer identification data, for example, name, address, phone number
- Account identification data, for example, account holder, account number, account type
- Library catalog information for physical books, videos, CDs

In general, you classify as item those item types that you want to use to store attributes only.

Restriction: Item types that are classified as item are not supported by the provided Client for Windows or eClient.

Item type classification: Resource item: *Resource items* describe and provide a connection to content that is stored on the resource manager. Examples of item types that you might classify as resource item:

- Roster of videotaped seminars that users can view over the Internet or your intranet
- Auto insurance accident data, such as photos and scanned police reports
- Library catalog information for scanned, digitally stored journals

When your users find resource items, they can view or launch the referred-to content directly from that resource item.

Restriction: Item types that are classified as resource item are not supported by the provided Client for Windows or eClient.

Item type classification: Document: Content Manager supplies a data model implementation that you can use, called the *document model*. The document model is similar to other document management systems and to previous releases of ImagePlus[®] and Content Manager in that it supports multi-part documents with related content. For example, subsets of pages are in different parts with associated graphical annotations and notes.

Modeling your data with the supplied document model instead of creating a similar data model from scratch has the following advantages:

- You can use the client applications that Content Manager provides.
- The performance of your system is better, because of the performance enhancements explicitly built into Content Manager specifically for the document model implementation.
- Writing your own application is simpler, because many of the decisions you might have to make have already been made.

When you classify an item type as document, you specify that this item type adheres to the document model. Examples of item types that you might classify as document:

- A journal article
- A journal
- A folder
- An insurance policy

A document item type is not required to have associated parts, for example a folder or similar container that is metadata only. Remembering that the document model is a data model implementation, you can see that a document item type without associated parts is similar to an item type that is classified as item.

If a document item type does have associated parts, they are managed in a parts list, which is a hidden child component of the document item type. You create the document parts first and then associate them with a document item type in the New Item Type Definition window on the Document Management page, as shown in Figure 8 on page 18.

Part type	Access control list	Resource manager	Collection	Version
ICMBASE	DocRouteACL	BETHESDA	CBR.CLLCT001	No
ICMBASESTREAM	DocRouteACL	BETHESDA	CBR.CLLCT001	No
ICMBASETEXT	DocRouteACL	BETHESDA	CBR.CLLCT001	No

Figure 8. Associating document part types with a document item type. You associate document parts with a document on the Document Management page in the New Item Type Definition notebook.

Requirement: Although a document item type is not required to have associated parts, a document item type must have at least one associated base part, even if it is empty, to be displayed in the eClient.

Item type classification: Document part: The provided document model also includes an item type classification of document part. After you classify item types as document part, you then associate the document parts with a document item type. You can associate any given document part item type with only one document item type; you cannot reuse document part item types in multiple document item types.

You associate document parts with a document in the Define Document Management Relations window (Figure 9), which you reach by clicking **Add** on the Document Management page of the New Item Type Definition notebook.



The dialog box 'Define Document Management Relations' contains the following fields and options:

- Part type: ICMBASETEXT
- Access control list: DocRouteACL
- Resource manager: BETHESDA
- Collection: CBR.CLLCT001
- New version policy:
 - No
 - Yes
 - User choice

Buttons at the bottom: OK, Cancel, Apply, Help.

Figure 9. Define Document Management Relations window

When you associate the document parts with a document, you can select one of the five predefined document part item types:

ICMANNOTATION

Contains additions to, or commentary about, the main data; following the document metaphor, annotations include sticky notes, color highlights, stamps, and other graphical annotations in the text of a document.

These are the typical annotation parts from previous releases of Content Manager. Using the Client for Windows or the eClient, your users can create graphical annotations, which are viewed on top of the file or document being displayed. Most client applications can show or hide these annotations.

ICMBASE

Contains the fundamental content of a document item type that stores any non-textual type of content, including image and audio.

Requirement: To be viewable in the eClient, all document item types must include at least one base document part.

ICMBASETEXT

Contains the fundamental content of a document item type that stores text content. If you plan to index a text part of your document, you should store the part in this part item type. Indexing a text part allows text search to be performed on the content of the part.

ICMNOTELOG

Contains a log of information entered by users. For example, indicating the reason that the insurance application was denied or instructions to the next reviewer of the document.

These are the typical notelog parts from previous releases of Content Manager. Using the Client for Windows or the eClient, your users can create, view, and edit notelog parts. Notelog parts contain the user ID, timestamp, and text comments as entered by client users.

ICMBASESTREAM

Contains streamed data, such as video.

Versions

In Content Manager, you can keep multiple versions of items and objects. When you create an item type, you can specify the versions for items of that type on the Definition page of the New Item Type Definition notebook. You can set one of the following version policies:

Always create

Creates a new version of the item whenever it is updated. Client users are unaware that additional versions are being created until the next time that they retrieve the item.

Never create

Updates a single stored item every time.

Prompt to create

Allows client users to decide whether they want to create a new version when they are updating an item.

If you set the version policy to allow multiple versions, you can set a maximum number of versions or allow an unlimited number. If you set a maximum number, when the specified maximum is reached, the oldest saved version is deleted automatically to save the next version.

The version policy that you set on the Definition page applies to attribute values. For example, if you set the version policy to allow multiple versions of items, then a user might change the value of the Surname attribute from Sanchez to Garcia and thus create a new, updated version of the item.

If the item type that you are creating is classified as a resource item or document part, the version policy applies also to the object on the resource manager.

If the item type that you are creating is a document, you can specify supplemental version policy information for the specific document parts. You specify this in the Define Document Management Relations window (Figure 9 on page 18), which you reach from the Document Management page.

You can set one of the following version policies specifically for the document parts:

- No** Does not allow multiple versions of the selected document part.
- Yes** Create a version of the selected document part whenever that object is edited.

User choice

The client user decides whether to update the version they are editing or store the updates in a new version.

The version policy for the document part supplements the version policy that you set on the Definition page. For example, on the Definition page, you might allow a maximum of three multiple versions. In the Define Document Management Relations window, you might specify **No** for the base part, but **Yes** for the notelog and annotation parts. In this case, one version of the base part and up to three versions each of the notelog and annotation parts can exist at any given time.

In the document model, versioning is specified at the document level and at the part level. If both versioning for the document and part are on, and if you create a new version of the part, a new version of the document is created. If parts are merely replaced (no new version of the part is created) and attributes are not changed, a new version of the document is not created.

Item type subset

An *item type subset* is a view of an item type that shows a specified set of data (a subset) that is included in items of that item type. For example, you might create an item type to use for employee data. You might want certain employees to be able to view different portions of that data. For example, all employees might be able to access an employee's location and phone number, but only the employee's manager can access the employee's salary history. The regular employees and the managers are using different item type subsets to view the information that they have access to and that is of interest to them.

In the Client for Windows, as in earlier Content Manager versions, the item type subset is called the *item type view* or *view*. Client for Windows users can see the views that they have access to on the Views page of the Preferences notebook.

In the underlying database, the item type subset is a view of database table columns. In Content Manager Version 8, you can provide an attribute value to filter the rows. With item type subsets, you can filter both the attributes and the rows of items that are available in an item type. **Important:** There can be only one filter per component type and the filter condition can only be set to equality. If a component is filtered at one level, levels below that level are filtered as well, but not levels above it. There is a performance impact for using row-based filters, especially when performing complex queries that access several component types that have row filters.

Restriction: When defining an item type subset for a hierarchical item type, you cannot skip a component level. For example, if you have a root component, child component, and grandchild component, in order for your item type subset to include information from the root and grandchild, it must also include at least one attribute from the child component.

Items

An *item* is a generic term for an instance of any item type, regardless of item type classification. For example, you might have item types called Insurance claim and Policy holder. Each individual claim that you create and each individual policy holder that you identify is generically referred to as an item.

Depending on the item type classification that you selected when you created the item type, the item can be:

- An item, which is self-contained and does not describe or represent an object on the resource manager. An item contains information that does not directly equate with an object. For example, if you look up a broad subject keyword, the resulting item might actually be a list of items that further narrow the subject or simply a long textual explanation.
- A resource item, which describes and connects to an object on the resource manager. If an object is a discrete piece of digital content, an item is a representation of that object. The item is not the object, but it thoroughly identifies the object and how to find it.
- A document or a document part, each of which is an element of the document model. For more information about the document model, see “Item type classification: Document” on page 17. (The system recognizes a document as an item and a document part as a resource item.)

Semantic types

The *semantic type* is a descriptive attribute for an item that helps applications to identify the behavior (semantics) for that item. Client applications use the semantic type to distinguish the use and purpose of different items. For example, you might use a document item type to store a document and another document item type to store a folder. The semantic type distinguishes the document from the folder.

You specify the semantic type when you create an item, and the semantic type is stored as an attribute value. You can select one of the following seven predefined semantic types:

Annotation

Additions to, or commentary about, the main data; following the document metaphor, annotations include sticky notes, color highlights, stamps, and other graphical annotations on a document.

Base The fundamental content of an item that stores any type of content, including image, text, and audio.

Container

A generic container for other items.

Document

A document, usually containing one or more base (ICMBASE) parts and possibly an annotation (ICMANNOTATION) and a notelog (ICMNOTELOG) part.

Folder A folder for containing items or other folders.

History

A log of activities for the associated item, entered as text by the application. This semantic type is available only for migration from earlier Content Manager versions.

Note A log of information entered by users. For example, indicating the reason that the insurance application was denied or instructions to the next reviewer of the document.

In addition to the seven predefined semantic types, you can create your own semantic types in your application.

Forming relationships between items

Restriction: Most of the function described in this section is not supported by the Client for Windows or the eClient. For a complete list of what is supported by the provided clients, see Table 3 on page 9.

This section describes the various ways that you can form relationships between items in Content Manager. Content Manager provides links and references, and the underlying relational database, DB2 Universal Database™, provides foreign keys. Table 4 summarizes the linking mechanisms.

Table 4. Advantages and restrictions of linking mechanisms

Linking mechanism	Used at component level:	Linked elements can be deleted	Limited by version?
Link	Root to root	Yes	No
Reference	Root or child to root	Specify when you create reference	Specify when you create reference
Foreign key	Root to a different item type or external table	Specify when you create the foreign key	Specify when you create the foreign key

Links

A *link* is a directional relationship at the root component level between two items: the source item and the target item. You can use links to associate one or more items with each other at the root component level at run-time. For example, assume that you have a Customer item and an Underwriter item, and you want to associate the two. Instead of making Underwriter a child component of Customer, you can associate the two by using a link.

In the system, you define a link, and the APIs create an entry in the links table to link the two items, as shown in Figure 10.

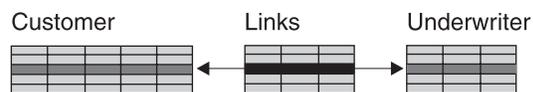


Figure 10. A link in action. Customer and Underwriter are root components of separate items; they are linked with a link that is specified in the links table.

As illustrated in the figure, the link is separate from the linked items. It is in a links table that contains information about which linked item is the source, which is the target, and the type of link.

Content Manager provides two link types: folder contains (DKFolder) and containment relationship (Contains). You can use the folder contains link type to mimic the connection of a physical folder and a contained document. In the New Link Type window shown in Figure 11, you can specify your own link types to symbolically represent the various links that are required for your data model. For the example shown in Figure 10 on page 22, you might want to use a link that doesn't imply containment, so you might create your own simple connection link.



Figure 11. Specifying a link type in the New Link Type window

You can link only between root components of different items. As summarized in Table 4 on page 22, there are no restrictions on links, other than privileges; either the source or target can be deleted. The link is independent of versions.

Content Manager also provides auto-linking. (Earlier Content Manager versions included a more restricted implementation of auto-linking called auto-folding; the implementation was restricted to folder linking only.) As shown in Figure 12 on page 24, you establish auto-linking when you create item types to automatically link related item types. You cannot establish auto-linking with an item type that does not exist.

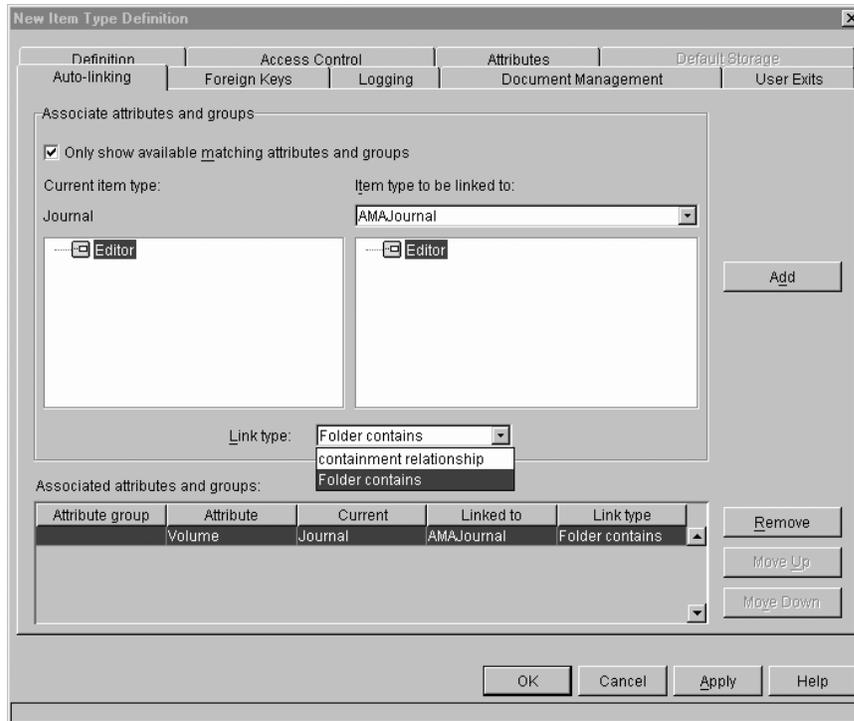


Figure 12. Specifying auto-linking. You specify auto-linking on the Auto-linking page in the New Item Type Definition notebook.

As with regular links, the auto-link is at the root component level. Any items that are created using the specified item types are automatically linked. If an item of one of the auto-linked types does not exist, it is automatically created, for example, if you create a form that must auto-link with a folder that does not yet exist, the folder item is automatically created.

When using the Folder Contains link type for auto-linking, add the auto-link rule to the item type that is the "content" of the folder. Set the **Linked to** field to the item type of the intended folder.

References

A *reference* is a single-direction, one-to-one association between a root or child component of an item and a root component of another item of the same or different item type. For example, assume that you have a Personal auto policy root component with an Insured vehicles child component and an Operators child component. You also have an Underwriter root component that you want to associate with certain Claims that are under the Customer root component. In Content Manager you can associate the Claims child component with the Underwriter root component by using a reference, which is shown as the arrow in Figure 13.

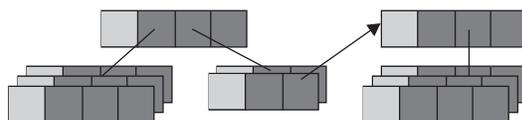


Figure 13. A reference in action

In the system, you define the reference as an attribute that is part of the source item.

When you create a reference, you supply a name and a display name (Figure 14).

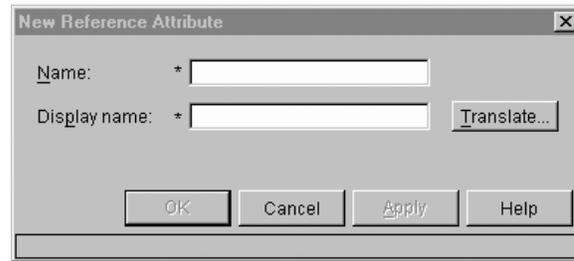


Figure 14. Creating a reference in the New Reference Attribute window

The reference is actually an attribute group, which includes system-defined attributes that define the connection.

You can create a reference to associate a root or child component of one item to the root component of another item. Table 4 on page 22 shows that when you create the reference, you can determine whether the target can be deleted if there is any reference to it.

Foreign keys

Foreign keys are supplied by DB2 Universal Database, the underlying database management system. A *foreign key* is a column or a set of columns in a table that refer to a unique key or the primary key of the same or another table. A *unique key* is a column or a set of columns for which no values in a row are duplicated in any other row. You can define one unique key as the *primary key* for the table. Each table can have only one primary key.

You use a foreign key to establish a relationship with a unique key or the primary key to enforce referential integrity among tables. In Content Manager, you can define foreign keys to another item type or to a database table that is not part of the Content Manager system. For example, you might have a database table that contains salary information. The database table is not part of the Content Manager system, but you do have an item type in Content Manager for employee data. You can create a connection between the employee data item type and the salary information table with a foreign key.

When you create an item type, you define foreign keys by clicking **Add** on the Foreign Keys page of the New Item Type Definition notebook. The Define Foreign Key window shown in Figure 15 on page 26 opens.

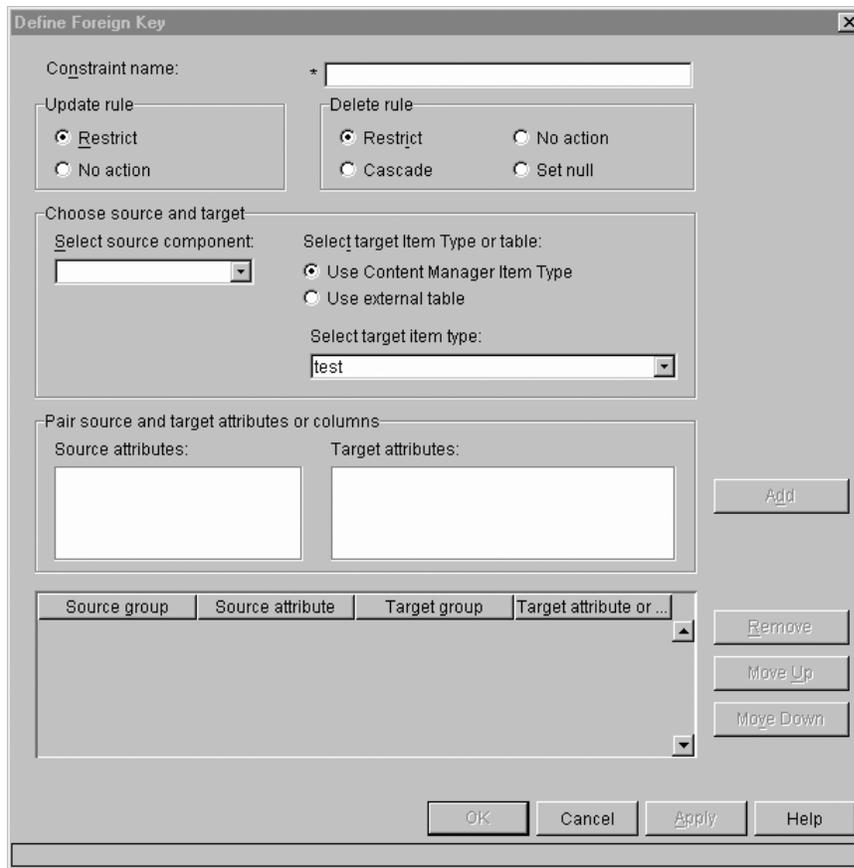


Figure 15. Define Foreign Key window

Objects

In Content Manager, an *object* is any data entity that is stored on a resource manager in digital form. Objects can include JPEG images, MP3 audio, AVI video, and a plain text file. For example, a few of the formats that are supported natively by Content Manager are: Microsoft Word, Lotus[®] WordPro, TIFF, and JPEG.

Objects are managed by items on the library server. The items contain the necessary information for describing and locating the objects. Using the items, users can create, retrieve, update, or delete objects.

MIME type

MIME type is an Internet standard for identifying the type of object that is being transferred across the Internet. MIME types include many variants of text, audio, image, and video data.

In Content Manager, when you create an object, you specify its MIME type. When an object of that type is retrieved from the resource manager, your application reads the MIME type and determines how to handle the object. For example, if the MIME type for an object is GIF, your application might launch a Web browser to view the object.

MIME type replaces content class from earlier Content Manager versions.

To properly handle the various types of data in Content Manager, each object needs to be associated with a MIME type (Multipurpose Internet Mail Extensions). Viewers need to know the MIME types for viewing certain documents. You must decide which data types Content Manager can use by identifying them to the system.

Content Manager ships with some predefined MIME Types, which you can view in the system administration client, that a client application can use. If you need to store data types not identified by the predefined MIME types, you have to add new ones. When you define a new MIME Type, then you need to use the following naming convention: content type/subtype.

A content type describes the contents of a document and allows the application to identify which view to use to present the document. A subtype specifies a specific format for the document. For example, the MIME type, image/jpeg, describes a file as being a image file while the subtype identifies that file as being of JPEG format. Available content types include, but are not limited to:

audio Audio files like music or voice recordings. Examples include: audio/basic and audio/mpeg.

application

Binary files and specific applications like Lotus Wordpro (application/vnd.lotus-wordpro) or Lotus Freelance (application/vnd.lotus-freelance).

image Image files like photos and drawings. Examples include: image/tiff and image/g3fax.

text Text files that can handle several character sets in several languages like HTML and XML files. Examples include: text/plain and text/html.

video Video or animated files like MPEGs. Examples include: video/mpeg and video/quicktime.

If you need to construct a MIME type that is not a standard MIME type, then you can define it using the naming convention: content type/x-subtype, where subtype is the user-specific subtype. For example, WAV files are not considered a standard MIME type, so, the MIME type name looks like this: audio/x-wav.

Important: if you define a MIME type that is considered a standard MIME type, and you use x-, the application you use might not recognize the document. For example, if you have an image that is a GIF, your browser can display it if you use the MIME type image/gif. However, if you define the MIME type as being image/x-gif, the browser does not recognize the subtype x-gif, and therefore, cannot display the image.

When you define a MIME type, you can also provide the usable suffixes for it. Suffixes assist MIME types to identify what type of data can be viewed on which viewer. However, most applications recognize file formats and identifies the appropriate viewer to view the MIME type, whether you specify a suffix or not.

To view the MIME types that have come with Content Manager, expand Data Modeling in the system administration client and click **MIME Types**. The right pane displays the predefined MIME types. If you want to define a MIME type, see the system administration client online help.

Media object class

The *media object class* describes the data that is contained in an object and how to act on it. When you create an object type, you specify its media object class. When an object of that type is retrieved from the resource manager, your application uses the specified media object class to appropriately handle the object.

Content Manager provides the following four predefined media object classes:

DKLobICM

Represents an abstraction for a generic large object (LOB) that is stored on a resource manager and pointed to by an item on the library server. Use DKLobICM to add, retrieve, update, and delete generic resource manager objects. To work with more specific types of data, you can use one of the more specific subclasses of DKLobICM: DKStreamICM, DKTextICM, and DKVideoStreamICM.

Some MIME types are inherently streamable, and so are appropriate for use with the DKStreamICM and DKVideoStreamICM media object classes. Other MIME types are text-searchable and are appropriate for use with DKTextICM. All MIME types can be stored as DKLobICM.

DKStreamICM

Represents generic streamable data that is stored on a resource manager and pointed to by an item on the library server. Use this class to:

- Add, store, or update large streamable objects from external sources using protocols like FTP. The adding or storing of objects can be synchronous or asynchronous.
- Retrieve (synchronously or asynchronously) large streamable objects to external destinations.
- Specify where to begin and end streaming.
- Retrieve information about stream duration, rate, format, and group.

This class is actually a subclass of DKLobICM.

DKTextICM

Represents text data that is stored on a Content Manager Version 8 resource manager and pointed to by an item on the library server. You can make a DKTextICM object text searchable by indexing the content of the object.

This class is actually a subclass of DKLobICM.

DKVideoStreamICM

Represents streamable video data that is stored on a streaming server (in this case, IBM Content Manager VideoCharger™) resource manager and pointed to by an item on the library server.

Because the content of DKVideoStreamICM objects is often large, you should complete add, update, and retrieve operations through third-party servers using a standard protocol such as FTP. After you retrieve the item from the library server, you can use this media object class to initiate a session to stream the content between the video server and player.

This class is actually a subclass of DKLobICM and inherits its methods from the DKStreamICM class.

One other predefined media object class, DKImageICM, has been deprecated. For more information about these media object classes and how to use them in your application, see the online API reference.

In addition to the predefined media object classes, you can define your own media object classes in the Media Object (XDO) Class Properties window, which is shown in Figure 16.

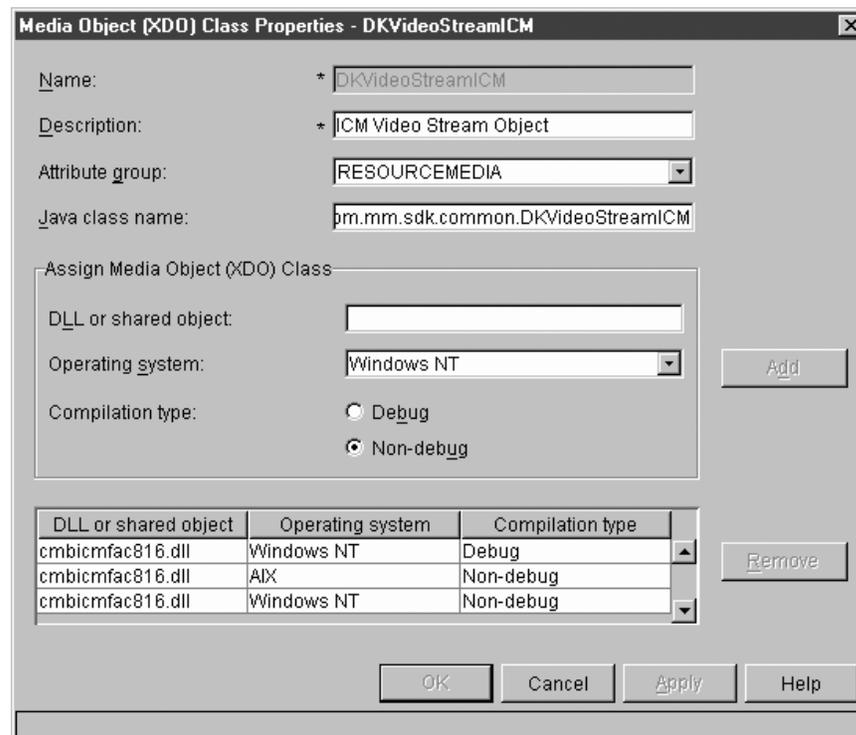


Figure 16. Media Object (XDO) Class Properties window

Text search

You can make attributes, resource items, and documents text-searchable from the system administration client. You enable each of these types of text search from the New Item Type Definition window.

You enable text search on the Definition page only for the Resource Item and Document item type classes. You enable attributes on the Attributes page. In the **Item type classification** field, you select **Resource item** or **Document** from the list and then select **Text searchable** to enable text search. You can use the default text search parameters or click the **Options** button to specify text search parameters on the Text Search Options window.

Text search uses the DB2[®] Version 7 Text Information Extender (TIE) or the DB2 Version 8 Net Search Extender. A detailed description of the text search parameters is described in the *IBM DB2 Text Information Extender Administration and User's Guide Version 7.2* or the *IBM DB2 Net Search Extender Guide Version 8.1*. The default text search settings are customized during TIE installation. To view the default settings, enter `db2 select * from db2ext.dbdefaults` in a DB2 command window for Windows or from any window in other systems.

After installing TIE, you will need to issue the following command to enable text search: `db2text enable database for text connect to <database name>`. Typically the database name will be `icm1sdb`. You must issue this command from a user ID with `sysadmin` authority for that database instance. If TIE is installed before

installing Content Manager, you can set the Content Manager install to automatically enable the database for text search.

Making documents text searchable

You can enable text search of the content of a document model. You do this by selecting **Document** in the **Item type classification** field and selecting **Text searchable**. You either specify a user-defined function on the Text Search Options page to fetch the content of the object or the default user-defined function is used.

You can make documents in popular formats such as Word and Word Pro[®] text searchable by specifying the user-defined function ICMfetchFILTER. You can also optionally add predefined Part types that are searchable.

Making attributes text searchable

You enable text search of attributes when you add attributes to an item type in the Attributes page. Each time that you add an attribute of type Character, Var Character, BLOB, CLOB, on the Attributes page for that item type, you have an option to make the contents of that attribute text searchable. To make the contents of the attribute text searchable select **Text searchable**.

You can use the default text search parameters or click the **Options** button to specify text search parameters on the Text Search Options window. If the item type contained an attribute for customer's last name, for example, then the user can query for that last name in a text search using a client application.

Making objects text searchable

You can enable text search of the contents of objects on the resource manager. You do this by selecting **Resource item** in the **Item type classification** field and selecting **Text searchable**. Either you specify a user-defined function on the Text Search Options page to fetch the content of the object or you use the default user-defined function.

Defining text search options

You can specify the text search parameters in the system administration client by clicking the **Options** button on the New Item Type Definition page or Attributes Page. The Text Search Options window opens. If you do not specify these parameters, default parameters are used.

In the **Index language settings** fields, specify a supported codepage (CCSID) and language code that is used to create the text index.

In the **Index update settings** fields, specify parameters to control the frequency that the index is updated. Specifically, you can specify the number of changes to the index before the next update, the amount of time that passes before the update. Leave the **Commit Count** field blank. Setting it to a non-zero value might lead to performance degradation.

Before you commit a change to the database, the database records a log file of changes that can be undone. When you commit the update, this log file is erased, making your updates to the database permanent. We currently recommend that you do not set a commit count to commit updates to the database. See the TIE documentation for further information about this situation.

In the **Storage options** fields, specify the directories on the library server where index and temporary files are stored.

In the **User defined function** fields, specify a user-defined function that allows text searching of resource items or documents.

Finally, in the **Model definition** fields, specify parameters for a model that might describe what sections of the text are to be indexed. The model consists of the name, a model file and the CCSID of the contents of the file. The model type is defined by the **Format** selection at the top of the Options window.

These parameters are described in more detail in the *IBM DB2 Text Information Extender Administration and User's Guide Version 7.2* or the *Net Search Extender Guide Version 8.1*.

Updating and reorganizing the index

The IBM DB2 Text Information Extender Administration and User's Guide Version 7.2 or the *Net Search Extender Guide Version 8.1* provides more detailed information about how to update and reorganize the index.

Content Manager includes a sample program that will update and reorganize the index for you. There are Java™ and C++ versions of the program with file extensions `.java` and `.cpp`, respectively. The name of the program is `STextIndexUpdateICM`. The method to call this application is in the opening lines of code. If you prefer, you can manually update and reorganize the index with the following procedure.

While you can use the **Index update settings** to control the frequency that the text index is updated, there are times when items are in a queue waiting to be updated. You can use the following command to immediately update the index:

```
Db2text UPDATE INDEX myindex FOR TEXT CONNECT TO icm1sdb USER icmadmin  
USING password
```

where:

- `myindex` is the name of the index. If you're unsure of the index name, you can find out by entering `db2 select indexname from db2ext.textcolumns`.
- `icm1sdb` is the name of the default database. You will need to substitute the database name if you renamed it.
- `icmadmin` and `password` are the user ID and password for the Content Manager administrator.

This command is useful when you have added several items to the system administration database and want to search them immediately.

If a text column is often updated, subsequent updates to the index can become inefficient. You can reorganize the index to improve performance. You can do this by entering the following command:

```
db2text update index myindex for text reorganize connect to icm1sdb user icmadmin using password
```

where:

- `myindex` is the name of the index. If you're unsure of the index name, you can find out by entering `db2 select indexname from db2ext.textcolumns`.
- `icm1sdb` is the name of the default database. You will need to substitute the database name if you renamed it.
- `icmadmin` and `password` are the user ID and password for the Content Manager administrator.

Modeling sample data structures

In this section, two scenarios describe how to model data in different situations. The first is a very simple scenario that describes the modeling of an article for publication in a journal. The purpose is to introduce how child components, links, and reference attributes can be used. The second scenario is insurance-related and is meant to be more realistic and complex. An automobile insurance policy is first discussed in simple terms. Then, different methods to model the data are presented in practical terms, including a discussion of reference attributes, folders, and links.

Scenario 1: Applying the building blocks

The data model building blocks and concepts that are applied here to the modeling of an article for publication in a journal.

An article is described by attributes such as Title, Date, and Author. This can be represented as a simple item type (see Figure 17) with one component type, which is called the root component.

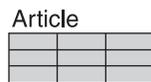


Figure 17. Simple item type

In a content management system, locating information can be simplified by associating a set of keywords with the document. These keywords, known as attributes, can have multiple values. Because you have multiple values, it is advisable to create a child component. In Figure 18, the third article in the Article item type has four keywords. Other articles can have different numbers of keywords.

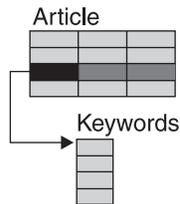


Figure 18. Item type with child component

Articles have one or more authors, as shown in Figure 19. You can define a second child component called authors, with attributes like name, company, and title.

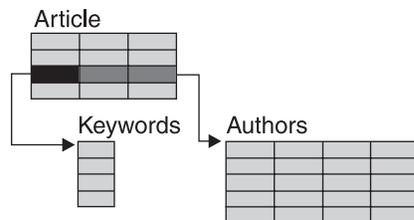


Figure 19. Item type with two child components

Although not likely, consider the case where Authors might have multiple addresses. Again, a child component can be used. In Figure 20, the third article has five authors, and the third author has two addresses.

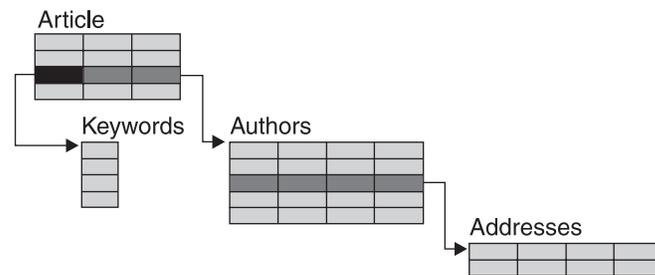


Figure 20. Item type with multiple child components

One problem here is that author information is duplicated. If the same author contributed to each of the four articles, four copies of the author and address records are needed. To eliminate duplication of data, you can create a separate item type called Authors and create a relationship between Articles and Authors.

The simplest and most common relationship between items is implemented by using links. The links table contains the source and target itemIDs, and the type of link. Figure 21 shows how you might use the folder contains link type, DKFolder, to mimic the connection of documents (articles) that are contained in a folder (journal). The links table contains the list of IDs for the folder and for the content within the folder. When you use linking, your application must provide referential integrity. Otherwise, journals can be deleted even if they contain articles.

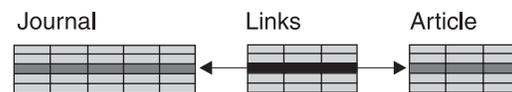


Figure 21. Example of linking

To create a relationship between either an item or a child component and another item, and to ensure referential integrity, you can use a reference attribute group. A reference is stored in the source component, either root or child, and consists of the target item ID, item type, component ID, component type, and version. In Figure 22, a child component named AuthorRef is created where each row contains a reference to an author. With this approach, any number of articles, books, or other components can reference a single Author record.

Reference attributes can be displayed in the eClient.

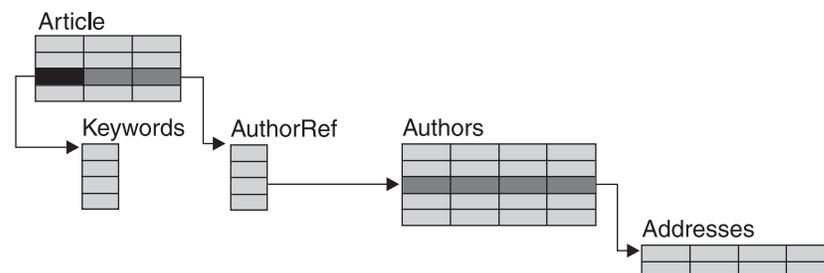


Figure 22. Example of reference attribute

Scenario 2: Modeling automobile insurance data

An automobile insurance policy contains information about both the policy holder and the policy itself. For example, policy holder information includes the name, address, and phone number of the customer. The policy is defined by a policy number, the description of the vehicle, including the vehicle identification number (VIN) and vehicle type, deductibles for comprehensive and collision loss, driver discounts, and so forth. Some of this information has a fixed number of values, whereas other information has a variable number of values. Each automobile policy has one policy number; however, different policy holders can differ in the number and type of discounts that they receive. A sample automobile insurance form is shown in Figure 23.

 XYZ Insurance Company 442 Main Street Gladville, OH 44555						State Vehicle number Policy Number OH 1MZ3872649VM OH57839657		Policy Period Effective May 26, 2002 to Aug 15, 2002			
Insured name and address Jane Smith 321 Poplar Drive Gladville, OH 44555						Operators Jane Smith Joe Smith					
Description of Vehicle(s)											
VEH	YEAR	MAKE	MODEL	BODY TYPE	MILEAGE	IDENTIFICATION NUMBER	VEH Use*				
02	02	Saturn	SL2	4D Sedan	12,540	1MZ3872649VM	8	15	15		
This location where the vehicle(s) is garaged is: (VEH 01) 321 Poplar Lane, Gladville, OH						*B=Business, W=Work, F=Farm, R=Recreation, S= School					
This policy provides ONLY the following coverages with related pricing noted.						VEH D=DED Premium Amount		VEH D=DED Premium Amount		VEH D=DED Premium Amount	
Part I - Liability Injury Option 1 \$ 100,000 Option 2 \$ 300,000 Option 3 \$ 25,000											
Part III - Uninsured Motorist Option 1/w deductible \$100,000 Option 2/w/o deductible \$300,000 Option 3 \$500,000											
Part IV - Physical Damage Coverage Comprehensive loss Collision loss Rental reimbursement Towing & Labor											
Total premium per vehicle: (For more detailed information, see the attached pages.)						752.47					
Discounts per vehicle: Anti-Theft discount \$ 9.65 Good Driver \$ 80.95 Air Bags \$ 10.45											
VIN: 1MZ3872649VM											

Figure 23. Sample automobile insurance form

You can use different methods to model this type of data. Consider the situation in which you create one item type called Policy Holder, as shown in Figure 24. This item type contains attributes such as name, address, and phone number. If this is the only item type that is defined, this model is not a good one because it does not include any content about the policy. It is merely a record containing information about customers with whom a company is doing business.

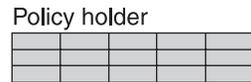


Figure 24. Policy Holder item type, with no content about the policy itself

You could create one item type called Automobile Policy, as shown in Figure 25. The root component might contain attributes such as policy number, those that describe the policy holder such as name, address, and phone number, and those that describe the policy such as VIN and vehicle type.

You can create a child component for this item type called Discount code. Because multiple values exist for discount codes (a customer can typically have more than one), a child component is a good place to include this type of information. Although this model does contain information about both the policy holder and policy itself, it is not the best model because of the problem of duplication of information.

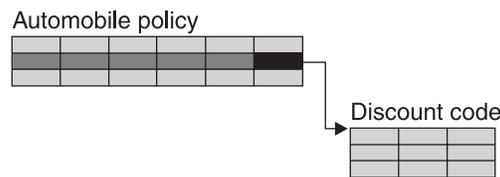


Figure 25. Automobile Policy item type with child component

Consider the situation in which a customer owns more than one car. A separate policy number exists for each car that the customer owns. If three policy numbers exist for a policy holder, three copies of the policy holder's address and phone number exist.

To eliminate the problem of duplication, you can create two item types: Policy holder (with attributes such as name, address, and phone number) and Automobile policy. Instead of putting an address attribute in the Automobile policy item type, you can create a reference attribute that you use to point to the Policy holder item type, as shown in Figure 26.

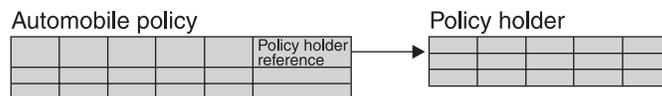


Figure 26. Automobile policy item type with reference attribute

Using the system administration client, you create a reference attribute called PolicyHolder in the New Reference Attribute window. On the Attributes page of the New Item Type Definition notebook for the Automobile policy item type, you can associate this reference attribute with this item type.

One potential advantage of reference attributes is referential integrity. If you select the **Restrict** delete rule on the Attributes page, you can prevent a policy holder from being deleted when a policy still exists.

Customers might have more than one type of policy. For example, they might have auto insurance, homeowners insurance, and life insurance. Another way to use child components is to create an item type called Policy holder that has a child component called Policy. The Policy child component might contain a reference attribute that is used to point to an item in the Automobile policy, Home policy, or Life policy item type. These three item types then contain attributes that describe them. The cardinality of the child component determines how many policies a customer can have.

Another method that you can use to build a relationship between item types is linking, as shown in Figure 27. Using the system administration client, you create the Policy holder item type and classify it as a document item type. The Policy holder folders link to items from other item types, such as Automobile policy and Home policy, which contain information about these specific policies.



Figure 27. Linking the Policy Holder folder with the Automobile Policy document

The Content Manager client applications allow documents or folders to be linked to folders. These items are not stored in a single place and contained within the folders as in a file system, but are linked to the folders. Documents and folders can be linked to multiple folders, whereas documents and folders typically are in one place in a file system. Using the eClient and Client for Windows, users can paste documents or add them into folders, which automatically creates the link.

Document item types generally consist of multiple document parts. With the system administration client, you can associate document parts with document item types on the Document Management page.

The Content Manager client applications require that each document item type has a base part. Typically, document item types have ICMBASE (base part), ICMANNOTATION (graphical annotations that overlay the base part), and ICMNOTELOG (separate textual comments).

The main content of an item in a document item type is stored as a base part. For example, the scanned picture of a car or insurance policy is the base part of an item in the Automobile policy item type. This item might then be added to a folder in the Policy holder item type, creating a link between the Automobile policy item and the Policy holder folder.

You can automatically populate folders by setting up auto-linking. Using the system administration client, open the folder item type and, on the Auto-linking page of the New Item Type Definition notebook, add a link to the document item type using the **Folder contains** link type. The advantage of auto-linking is that the system automatically places any document that you create in the client into the folder.

You can use foreign keys for validation purposes. You use them to establish a relationship with a unique or primary key to enforce referential integrity among

tables. For instance, in a Policy holder item type, you can create a unique attribute called customer number. When you create an Automobile policy item type, that item type might also have the customer number attribute. You can then define a foreign key, using the Define Foreign Key window. The foreign key points to the customer numbers in the Policy holder item type so that you cannot enter an incorrect customer number when you enter data for the automobile policy.

Modeling your data

This section describes how to identify and model your data in Content Manager. Each of the following steps is described in detail:

1. Identify your data.
2. Separate your data into operational and non-operational.
3. Sort your data into like types.
4. Identify your users and what data they need to access.
5. Within each data type, identify elements that might be searched for.
6. Identify hierarchies and elements that might have multiple values.
7. Diagram data relationships.
8. Decide whether you require a custom data model.
9. Model your data in Content Manager.

XYZ Insurance is a fictitious insurance company that is introduced in *Planning and Installing Your Content Manager System* and is used throughout this section. Each step description ends with an example of what XYZ Insurance does to complete that step.

Step 1: Identify your data

To begin modeling your data in Content Manager, you must first identify your data. Identifying all of the data that you want to include in the system helps you see the relationships among the data and the needs of your business. This process also exposes the requirements for your data model.

To begin integrating Content Manager into your business, you might decide to begin by using it for a certain area of your business. Try to select an area that is self-contained, so that you do not need to significantly alter your model later as you add new areas.

At first, don't label or judge the data that you collect, just identify and list it. Examples of data (either online or printed) that you might list are:

- Forms
- Documents
- Photos
- Videos
- Graphics
- Presentations
- Audio

To identify your data, you can try any or all of the following methods with a worksheet like that shown in Table 5 on page 38:

Analyze your business procedures

Determine what procedures and processes your business regularly follows.

Are forms, documents, or other objects required throughout these procedures and processes? Do any online forms or repositories require data entry during a procedure? Is there data, stored online or in printed format, that is an input to any step in the process?

On your worksheet, list each of these documents, forms, and data by a recognizable name. Do not worry about the order of the elements that you list. If you know who uses the elements that you list, you can indicate those names or job titles in the second column.

Identify the roles in your business

List the roles of the employees in your business and determine what each of them needs to do their jobs. You might even interview or observe representatives of different roles to see what they do and what they use to do it.

Identifying the roles, and needs of each, is especially helpful if you want to use Content Manager to automatically route documents through a process. Identifying roles is also a good way to find data that should be modeled in the system but that does not fit into a recognizable business procedure or process, such as educational materials.

On your worksheet, list all of the documents, forms, and reference data that are used by each representative role in your business. List these elements by recognizable names and identify the roles who need them. If these documents, forms, or data pass through a process that you want to model in a specific order, you should indicate the order on your worksheet.

Identify your data resources

In addition to data that is used during day-to-day business, most companies have data that is used infrequently. An example of such data is materials that are used for classes or training sessions. On your worksheet, list all of this resource data that you want to include in your system.

Table 5. Sample worksheet 1, columns 1 and 2. Use these columns to identify your data and who uses it.

Document, form, element of data	Used by	Reserved for later steps

XYZ Insurance uses a combination of analyzing its business procedures and identifying the roles in its company to identify their data. Table 6 shows some of the data that XYZ Insurance identifies.

Table 6. XYZ Insurance completes sample worksheet 1, columns 1 and 2

Document, form, element of data	Used by	Reserved for later steps
Personal auto policy	Agent, underwriter	
Homeowners policy	Agent, underwriter	
Auto claim form	Agent, claims adjuster, underwriter, accounts payable	
Damage photos	Claims adjuster	

Table 6. XYZ Insurance completes sample worksheet 1, columns 1 and 2 (continued)

Document, form, element of data	Used by	Reserved for later steps
Police reports	Claims adjuster	
Training manual	Underwriter	
List of approved defensive driving courses	Agent	

Step 2: Separate your data into operational and non-operational

In this step, you examine the list of data that you identified in “Step 1: Identify your data” on page 37, and you identify which data is operational and which data is non-operational.

Operational data is the data that you need to perform business procedures and processes, for example, an insurance policy or claim form. *Non-operational data* is the information that you use for reference, research, education, and so forth, for example, materials from a training session or a videotape of a session with the company president.

Separating your data like this can help you make decisions about how to use Content Manager effectively to model your data. The following list identifies some considerations that separating your data can help you with:

- Operational data might require workflow. You might decide to use Content Manager’s document routing function or EIP’s advanced workflow to create a routing system for operational data that follows a process, for example, a claim form that is passed from receiver to adjuster to approver to cashier.
- Operational data might require heavy client application usage. The clients that are provided by Content Manager do not support all of the elements that you can use to model your data (see Table 3 on page 9). If you want to use one of the provided clients, you must model your data accordingly. You need to make an informed decision about whether to model your data using the full function of Content Manager because doing so requires you to write your own client application.
- Non-operational data might not require the immediate performance expected of operational data.

Table 7 is an extension of the worksheet in Table 5 on page 38. One of the reserved columns is now labeled “Operational?” so that you can use it to indicate whether each element of data is operational or non-operational.

Table 7. Sample worksheet 1, column 3. Use this column to separate operational and non-operational data.

Document, form, element of data	Used by	Operational?	Reserved for next step

In Table 8, XYZ insurance separates the data that it identified earlier into operational and non-operational.

Table 8. XYZ Insurance completes sample worksheet 1, column 3

Document, form, element of data	Used by	Operational?	Reserved for next step
Personal auto policy	Agent, underwriter	Yes	
Homeowners policy	Agent, underwriter	Yes	
Auto claim form	Agent, claims adjuster, underwriter, accounts payable	Yes	
Damage photos	Claims adjuster	Yes	
Police reports	Claims adjuster	Yes	
Training manual	Underwriter	No	
List of approved defensive driving courses	Agent	No	

Step 3: Sort your data into like types

To complete this step, you examine and begin to make decisions about the data that you gathered. Sorting the data into like types helps you begin to develop a structure for your data model. After you complete this step, you will have a preliminary list of the item types that you want to create in Content Manager to model your data.

Begin this step by consolidating any duplications on your worksheet.

Examine your worksheet (see Table 9) and identify areas of commonality between elements that are listed in column 1. Use the full width of column 4 to try a combination of the following techniques for sorting the elements into like types. Sort by:

- Media type, for example, documents, videos, photographs, and so forth
- Paper forms
- Purpose
- Customer type

By using a combination of techniques, you can drill down to types that are unique and begin to uncover where unique information appears in more than one place. For example, you might sort by media type, identifying documents, videos, and photographs. You might then sort each by purpose, identifying these types of documents: insurance claim, personal auto policy, police report, fax, and so forth.

Table 9. Sample worksheet 1, column 4. Use this column to identify unique types.

Document, form, element of data	Used by	Operational?	Unique types

In Table 10, XYZ Insurance sorts the data that it gathered into unique types. First XYZ Insurance sorts the data by media type, identifying scanned documents, digital photos, an online source (Microsoft Word) document, and a plain text (ASCII) list that was stored in Wordpad on an agent’s desktop. The results of sorting by media type appear first in column 4 of the table.

Next, XYZ Insurance sorts by paper form, noting that the scanned documents are different enough to each require a unique type. The Damage photos and Police reports are to be associated directly with the Auto claim form. The Training manual and List of approved defensive driving courses are not related to any forms, and so are unique. However, other training manuals and lists of information might be used for reference, so these unique types should be generic enough to encompass that other data, too. The results of the second sorting pass appear second in column 4 of the table.

Table 10. XYZ Insurance completes sample worksheet 1, column 4

Document, form, element of data	Used by	Operational?	Unique types
Personal auto policy	Agent, underwriter	Yes	Scanned document; Personal auto policy form
Homeowners policy	Agent, underwriter	Yes	Scanned document; Homeowners policy form
Auto claim form	Agent, claims adjuster, underwriter, accounts payable	Yes	Scanned document; Auto claim form
Damage photos	Claims adjuster	Yes	Digital photo; Detailed information for Auto claim form
Police reports	Claims adjuster	Yes	Scanned document; Detailed information for Auto claim form
Training manual	Underwriter	No	Microsoft Word document; Manual not related to a form
List of approved defensive driving courses	Agent	No	ASCII text document; Reference list not related to a form

Step 4: Identify your users and what data they need to access

So far, you’ve focused primarily on identifying the data that you use and need in to run business. In this step, you identify who needs that data.

As part of building your content management system, you must identify your users and provide them with appropriate access control. Access control is a big topic, which is not covered in this document. (See the *System Administration Guide* for information about controlling access in your system.) However, identifying your users and what they need to access at a very basic level is an important step for building your data model. Knowing who needs what can help you determine how you use Content Manager effectively.

Obviously, when you build your system, you want to maximize performance. The provided clients are built to maximize performance, but they have some restrictions on the data that they display to users (Table 3 on page 9). For example,

after completing this step, you might realize that, although you have many users, they need access to a small subset of the data.

Look at your worksheet. If you have not already done so, use the second column to identify users (by role) for the different unique types that you identified. If you used the method of identifying business roles in “Step 1: Identify your data” on page 37 to identify your data, you have already begun to identify the users of your data. Even if you completed the second column earlier, look through it again, using the information that you entered into the fourth column.

Tip: Try to save some room in the second column so that you can plan your access control later.

XYZ Insurance completed the second column earlier. After reviewing the worksheet, XYZ Insurance realizes it wants to be able to print renewal policies directly from the system onto special forms, which they can send to customers. So, although customers do not need direct access to the system, they are indirect users of the system in the sense that the system must provide output that is tailored for their needs.

Table 11. XYZ Insurance updates sample worksheet 1, column 2

Document, form, element of data	Used by	Operational?	Unique types
Personal auto policy	Agent, underwriter, customer	Yes	Scanned document; Personal auto policy form
Homeowners policy	Agent, underwriter, customer	Yes	Scanned document; Homeowners policy form
Auto claim form	Agent, claims adjuster, underwriter, accounts payable	Yes	Scanned document; Auto claim form
Damage photos	Claims adjuster	Yes	Digital photo; Detailed information for Auto claim form
Police reports	Claims adjuster	Yes	Scanned document; Detailed information for Auto claim form
Training manual	Underwriter	No	Microsoft Word document; Manual not related to a form
List of approved defensive driving courses	Agent	No	ASCII text document; Reference list not related to a form

Step 5: Within each data type, identify the elements that might be searched for

In this step, you develop the unique types that you identified. For each unique type, you identify the characteristic elements, the attributes that users of your system might use to search for items. You must consider how you plan to use your system so that you can identify the right number of attributes to uniquely identify items of a given type.

You might decide to store few characteristic elements, just enough for users to search for and find items. For example, you might use the system to store scanned

documents that users can find by entering a customer name or customer number. In such a system, users review the scanned document to see the details. Or you might use the system to store all customer information in such a way that you can print customer documents onto pre-printed forms. In this type of system, you would define many attributes, and users could search for items by entering almost anything about a customer.

In a new worksheet, like sample worksheet 2 shown in Table 12, copy the unique types that you identified on your first worksheet into the first column. Then, use the second column to identify the necessary attributes. In the third column, make any notations regarding the data type, length, and so forth of the attributes; doing this can help you later when you enter the attributes into the system.

Table 12. Sample worksheet 2, columns 1, 2, and 3. Use these columns to identify and describe your attributes.

Unique types	Characteristic elements	Data type, length	Reserved for next step

Table 13 shows how XYZ Insurance identifies the characteristic elements for a couple of the unique types that they previously identified. Because XYZ Insurance wants to use the system to print policies on special forms, it must identify attributes for those forms that must conform to the special, pre-printed forms.

Table 13. XYZ Insurance completes sample worksheet 2, columns 1, 2, and 3

Unique types	Characteristic elements	Data type, length	Reserved for next step
Personal auto policy form	Policy number	Alphanumeric character, 10	
	Named insured	Variable character, 128	
	Named insured address	Variable character, 512	
	Agent name and address	Variable character, 1024	
	Policy period	Date	
	Insured vehicles	N/A	
	Operators	N/A	
Damage photo (detailed information for Auto claim form)	Policy number	Alphanumeric character, 10	
	Photo date	Date	
	Auto claim form number	Alphanumeric character, 8	
	Description	Variable character, 1024	
Reference list	Title	Variable character, 30	
	Description	Variable character, 1024	
	Date	Date	

Step 6: Identify hierarchies and elements that might have multiple values

You can use Content Manager to build a robust data model, for example, modeling data in a hierarchy, allowing attributes to have multiple values, or both. In this step, you examine your data from “Step 5: Within each data type, identify the

elements that might be searched for” on page 42 and identify any hierarchies and any elements that might have multiple values.

Multi-valued attributes represent the simplest situation that requires you to create a child component. Note that, unlike previous Content Manager releases, with child components, you can have sets of attributes that might require multiple values, for example, an address that is made up of Street, City, State, and PostalCode. By making this set of attributes a child component, you ensure that the specified multiple values remain consistent with each other. If you have two addresses, the Street for the first address remains with its associated City, State, and Postal Code, a situation that you could not guarantee if these multi-valued attributes were separated.

By completing this step, you expand your growing data model from identified item types and their attributes to include child components.

Table 14 is an extension of Sample worksheet 2. The reserved column is now labeled “Multiple values or child component” so that you can use it to identify attributes that can have multiple values or sets of attributes that should be moved into a child component.

Table 14. Sample worksheet 2, column 4. Use this column to identify attributes, or sets of attributes that might have multiple values. Also use the column to identify sets of attributes that you want to separate into a child component.

Unique types	Characteristic elements	Data type, length	Multiple values or child component

In Table 15 XYZ Insurance identifies the sets of attributes that require multiple values. A Personal auto policy can cover more than one vehicle and can include more than one operator (driver) who lives at the same address. XYZ Insurance wants to use child components for these sets of attributes.

Table 15. XYZ Insurance completes sample worksheet 2, column 4

Unique types	Characteristic elements	Data type, length	Multiple values or child component
Personal auto policy form	Policy number	Alphanumeric character, 10	No
	Named insured	Variable character, 128	No
	Named insured address	Variable character, 512	No
	Agent name and address	Variable character, 1024	No
	Policy period	Date	No
	Insured vehicles	N/A	Yes
	Operators	N/A	Yes
Damage photo (detailed information for Auto claim form)	Policy number	Alphanumeric character, 10	No
	Photo date	Date	No
	Auto claim form number	Alphanumeric character, 8	No
	Description	Variable character, 1024	No

Table 15. XYZ Insurance completes sample worksheet 2, column 4 (continued)

Unique types	Characteristic elements	Data type, length	Multiple values or child component
Reference list	Title	Variable character, 30	No
	Description	Variable character, 1024	No
	Date	Date	No

Step 7: Diagram data relationships

So far, the data that you gathered is a lot of words on two worksheets. You are probably aware of connections between different rows of the worksheets. By diagramming the data from the worksheets, you can get a more complete view of the model that you want to build, especially the links and references that relate different elements.

Review your second worksheet to identify and diagram the connections between root and child components (and child components to grandchildren, and so forth). Also diagram relationships between item types, and indicate whether these relationships are links or references. Look especially for situations where you have data this is used repeatedly. For example, if you have some “boilerplate” information that is included on all your forms, you might store that in a different item type and link to it from the other item types that use that information.

Figure 4 on page 14 shows the diagram that XYZ Insurance might draw for the Personal auto policy form with the Insured vehicles and Operators child components. XYZ Insurance also benefits from drawing a simple diagram showing how it wants to collect Auto claim form, Damage photo, and Police report into an Auto claim folder, and use folder links to connect the four item types.

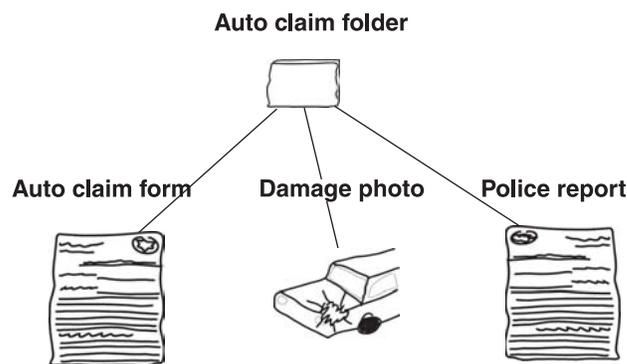


Figure 28. Simple relationship diagram. The Auto claim folder contains the Auto claim form, Damage photo, and Police report. The four item types are all document item types.

XYZ Insurance also identifies some basic customer information that it collects on most forms and does not want to repeat. XYZ Insurance separates these specific attributes into a separate type, called Customer data, which can be referenced from the various form item types.

Step 8: Decide whether you require a custom data model

This step is really a decision-making step. Consider the data that you have gathered and diagrammed and any other requirements for your system, so that you can determine the best way to use Content Manager to fit your needs. In previous steps, particularly “Step 2: Separate your data into operational and

non-operational” on page 39 and “Step 4: Identify your users and what data they need to access” on page 41, you gathered information that can help you with this step.

Content Manager provides an implementation of the data model called the document model (see “Item type classification: Document” on page 17). If you decide to use the document model to model your data, you can use the provided client applications (Client for Windows and eClient) or write your own application. If you elect to design a custom data model, you must write your own application.

The provided client applications have some limitations on the data that they display to users. For example, in “Step 6: Identify hierarchies and elements that might have multiple values” on page 43, did you identify a need for more than one level of child components? If so, client users will not be able to view those lower levels. See Table 3 on page 9 for a complete list of what the provided clients support.

XYZ Insurance reviews the data that it gathered and diagrammed. XYZ Insurance has a large number of users (customer service personnel) who must access the basic customer and insurance data for all policies and claims. These users require high performance.

XYZ Insurance *did* identify some basic customer data that it wanted to connect with various forms using references. References are not supported by the provided clients. Furthermore, XYZ Insurance determines that it can model the rest of the data using the document model with one child component level. XYZ Insurance decides to defer the separation of basic customer data because it needs to have a solution working quickly and because of the performance needs of its users.

XYZ Insurance has, however, also identified the critical requirement that it wants to use pre-printed forms to generate renewal policies directly from Content Manager. To do this effectively, XYZ Insurance decides to code a custom application.

Step 9: Model your data in Content Manager

In this step, you “convert” the data that you gathered and diagrammed in the previous steps into a Content Manager data model. You complete this step on paper so that when you’re ready to model the data in the system, you have all the information that you need available.

This step consists of two alternatives. If you plan to model your data using the provided document model, proceed to the next section. If you plan to model your data using a custom data model, go to “Model your data by creating a custom data model” on page 48.

Model your data using the supplied document model

You already gathered your data and used it to make decisions about how to model it in Content Manager. In this step, you fit your data into the supplied document model. Doing this step on paper, before you begin entering the data into Content Manager, helps you to enter the data more quickly and helps to avoid rework.

If you have enough room, you can use your two worksheets and diagrams to label your document and document part item types, your child components, and your folder links. Or, you can use a new worksheet, like that shown in Table 16 on page 47 to list and label this information in one place.

Table 16. Sample worksheet 3, document model

Document item types	Document part item types	Child components	Attributes	Linked to:

Table 17 shows how XYZ Insurance fits their gathered data into the document model. Note that XYZ Insurance decided to create an Auto claim folder, which is a document item type. The Auto claim folder uses a folder link to connect with the included document part item types: Auto claim form, Damage photo, and Police report.

Table 17. XYZ Insurance completes sample worksheet 3, document model

Document item types	Document part item types	Child components	Attributes	Linked or referenced to:
Personal auto policy form	Personal auto policy form base	--	See Table 14 on page 44	--
	--	Insured vehicles	<ul style="list-style-type: none"> • Year • Make • Model • Style • VIN 	--
	--	Operators	<ul style="list-style-type: none"> • Number • Name • Birth date • Sex • License number 	--
Homeowners policy form	Homeowners policy form base		<ul style="list-style-type: none"> • Policy number • Named insured • Named insured address • Agent name and address • Policy period • Covered property 	--
Auto claim folder			<ul style="list-style-type: none"> • Name • Description 	Folder link to: Auto claim form; Damage photo; Police report
	Auto claim folder notelog		...	
	Auto claim folder history		...	
Auto claim form	Auto claim form base		<ul style="list-style-type: none"> • Policy number • Named insured • Affected vehicle • Incident date • Damage description 	

Table 17. XYZ Insurance completes sample worksheet 3, document model (continued)

Document item types	Document part item types	Child components	Attributes	Linked or referenced to:
Damage photo	Damage photo base		See Table 14 on page 44	--
Police report	Police report base		<ul style="list-style-type: none"> • Report number • Accident date • Officer name 	--
Training manual	Training manual base		<ul style="list-style-type: none"> • Title • Description • Author/owner • Audience 	--
Reference list	Reference list base		See Table 14 on page 44	--

What's next?

- If you plan to use the provided clients, the next step is to use the worksheets with the *System Administration Guide* and system administration online help to model the data.

Performance tip: When you model your data in the system administration client, you might want to create an index of the attribute values used for finding items. The index is created, in sorted order, and managed by DB2. When users search for values, matches are identified with little required I/O, which provides good response time and minimizes server CPU and I/O time. You must balance the increased performance benefit during retrieval against the relative performance cost of maintaining the index, for example, if you index every attribute in every component, you can affect the performance time for creating items.

- If you plan to write your own application:
 - See the *Workstation Application Programming Guide* and online API reference (Javadoc) for specific information about writing your application.
 - See `SItemCreationICM.java` in the `cmbroot\samples\java\icm` directory for specific API information about coding an insurance application similar to the one that is described in this document. For a complete list of the samples that make up the insurance scenario, see the samples README file: `README_SAMPLES_JAVA_ICM.txt`.

Model your data by creating a custom data model

You already gathered your data and have used it to make decisions about how to model it in Content Manager. In this step, you fit your data into a model that you can enter into Content Manager, identifying the various building blocks for your elements. Doing this step on paper, before you begin entering the data into Content Manager, helps you to enter the data more quickly and helps to avoid rework as you shuffle elements to maximize performance and reuse.

If you have enough room, you can use your two worksheets and diagrams to label your item types, resource item types, child components, links, and references. Or, you can use a new worksheet, like that shown in Table 18 on page 49 to identify this information in one location.

Table 18. Sample worksheet 3, custom data model

Item types, classification: item	Item types, classification: resource item	Linked to:	Child components	Attributes	Referenced to:

What's next?

- See the *Workstation Application Programming Guide* and online API reference (Javadoc) for specific information about writing your application.
- See `SItemTypeCreationICM.java` in the `cmbroot\samples\java\icm` directory for specific API information about coding an insurance application similar to the one that is described in this document. For a complete list of the samples that make up the insurance scenario, see the samples README file: `README_SAMPLES_JAVA_ICM.txt`.
- Use the worksheets that you created throughout this section with the *System Administration Guide* and system administration online help to model the data.

Chapter 3. Defining and configuring servers

The library server stores, manages, and provides access control for content stored on one or more resource managers. The library server processes requests from one or more clients and maintains data integrity between all of the components in the Content Manager system. A single library server can support multiple resource managers and data can be stored on any of these resource managers.

The resource manager is the repository for content stored in the Content Manager system. Content is always associated with a specific collection on a resource manager. Access to the content is done through the library server.

To define, configure, or add servers, you need to have the following information for each server:

- Server name
- Server type
- Host name
- User name
- Access to the server (such as a valid user ID and password)
- Protocol
- Port
- Schema
- Path

After you collect this information, you can add any server to the system administration program or update the current server information.

Defining a library server

The system administration client allows you to manage multiple library servers without you having to log off of the current library server and log back on to another library server. You can check what library servers you have by looking at the `cmbsrvs.ini` file located in the `%CMCOMMON%` directory, or logging on to the system administration client.

If you want to define more library servers to your Content Manager system, then you need to use the Server Configuration utility, located by clicking **Start** → **Programs** → **IBM Content Manager for Multiplatforms V8.2** → **Server Configuration**. Click **Apply**, then **OK** after you have completed this panel. You can click **Cancel** if you want to enter the configuration information at a later time.

Connecting to a local and remote database

This section explains how to connect the system administration client to a local and a remote administration database.

Connecting the client to a local administration database

In this configuration, the client and administration database are installed on the same Windows server.

1. Click **Start** → **Programs** → **IBM Content Manager for Multiplatforms V8.2** → **System Administration Client**.

2. If you have multiple databases on the server, select a database.
3. Type the database administrator ID, or the database connect user ID and password defined when the database was installed.
4. Click **OK**.

The administration client window appears and the local database name is displayed in the left pane.

Configuring the client to a remote database

Prerequisite: Configuring the client to connect to a remote database requires that the client workstation has either DB2 or DB2 Client Configuration Assistant (CCA). If the client workstation does not have already installed, you must install DB2 Client Configuration Assistant (CCA). To install DB2 CCA:

1. Insert the DB2 installation CD-ROM in the client workstation CD-ROM drive.
2. From the Installation menu, click **Custom**.
3. Click **Administration**.
4. Click **Client Configuration Assistant**. Uncheck the other options.
5. Click **Next**.

Connecting the client to a remote database is a three-step process:

1. You catalog, or add, the remote database.
2. You modify the `cmbicmsrvs.ini`.
3. You logon to the remote database through the system administration client.

These steps are expanded in detail in "Add a remote database using DB2 CCA".

Add a remote database using DB2 CCA: To add a remote database:

1. Click **Start** → **Programs** → **IBM DB2** → **Client Configuration Assistant**.
2. Click **Add**.
3. Select Search the network.
4. Click **Next**.
5. Expand Known Systems. If you do not see the name of the system where the remote database is installed, click **Add System**.
6. In the Host name field, enter the server name where the database is installed.
7. Click **OK**. **Tip:** If this process fails, and you are cataloging an AIX® database, make sure the DB2 administration server on your AIX box is turned on.
8. Expand the tree of the system name under Known Systems.
9. Click on the name of the database you want to connect to.
10. Click **Next**.
11. Enter an alias name (up to eight characters).
12. Click **Finish**.
13. Click **Test Connection**. Enter the administration or DB2 connect user ID and password that were defined when the database was installed, then click **OK**.
14. A dialog box indicates the success or failure of the connection test.
15. Click **Close**.

Catalog the remote database using DB2 command line: To catalog a remote database:

1. Click **Start** → **Programs** → **IBM DB2** → **Command Line Processor**.
2. At the `db2=` prompt, type the following data on one line:

```
db2=> catalog tcpip node [xxx] remote [hostname<fully-qualified>]
server [50000]
```

```
db2=> catalog database [yyy] as [alias_name] at node xxx
```

```
db2 connect to [alias_name] user <administrator id defined during
database installation> using <administrator password defined
during database installation>
```

xxx can be any value. The port 50000 must be the port number of the database instance the server is running on. **Tip:** If you have problems cataloging an AIX database, check /etc/services to find the right port.

In the second command, yyy is the name that you want to use for the database on the server. The other database name [alias_name] is the database name on the remote machine, and can be any name (up to eight characters).

Add database information to .ini files: In this step, you modify the cmbicmsrvs.ini configuration file.

If the installer installed the configuration files on the client workstation:

1. Navigate to x:\Program Files\IBM\CMgmt.
2. Open cmbicmsrvs.ini in a text editor.
3. Copy and paste the existing text and customize the settings for the new database:

```
ICMSERVER=<database alias name defined when you added or cataloged
the database>
```

```
ICMSHEMA=<schema defined when database was installed>
```

```
ICMSSO=<Single Sign-On setting defined when database was installed>
```

```
ICMDBAUTH=<Client or server authorization setting defined
when database was installed>
```

4. Save cmbicmsrvs.ini.

Connecting to the remote database: To connect to the remote database:

1. Click **Start** → **Programs** → **IBM Content Manager for Multiplatforms V8.2** → **System Administration Client**.
2. Select the remote database name.
3. Type the database administrator ID, or the database connect user ID and password you used to catalog or add the database.
4. Click **OK**.

The administration client window appears and the remote database name is displayed in the left pane.

Configuring a library server

After you define a library server to the system administration client, you need to configure it.

Your task is to assign the resource managers to a library server, maintain the INI files, and define the languages that each library server will support for index information. Each library server can support the index information (attributes and item types) for objects in one or more languages.

You assign a default resource manager and collection to users when you create users. You assign a default resource manager and collection to an item type when you create an item type. You also assign the languages that are used for index information to attributes and item types when you create attributes and item types.

Allowing trusted logon

When you set up your library server to allow trusted logon, you let users have access to the library server using their workstation password and without prompting for an additional password.

You must complete three steps to allow for trusted logon:

1. On the Library Server Configuration Definition page, select **Allow trusted logon**.
2. Assign the privilege set UserDB2TrustedConnect to the Connect User ID.
3. CM users can now log on to the library server without providing a password.

Adding a resource manager to the library server

When you add a resource manager to a library server, you need to have the resource manager's server name, host name, operating system, protocol, port, schema, path, and a system administrator's user ID and password.

The host name that you specify is sent to clients that need to communicate with the resource manager. If the server that your clients are trying to access is located on a private network and must be accessed from the Internet, then you should use a fully qualified domain name server (DNS), for example, `hostname.mycompany.com`. If your networks do not use or have domain name servers, then you should specify an IP address (for example, `9.87.65.432`) to ensure that all clients can locate the server.

To add a resource manager to a library server, you need to define the name of the resource manager, its hostname, platform, token duration, its access types and whether you need to enable the LAN cache. By enabling the LAN cache, you are providing a caching area for the resource manager to access when it needs to retrieve items for a client application request.

Adding a resource manager to the library server:

1. In the System Administration Client window, right-click **Resource Managers**.
2. Click **New**. The New Resource Manager Definition window appears.
3. Enter your resource manager information and click **OK**.

The resource manager name now appears in the list beneath the Resource Managers node in the system administration client main window.

Next, you need to configure the SMS components.

Changing the library server and system administrator's password to the resource manager

If you need to change the password to the resource manager, then you need to change the password for the log on of the library server to the resource manager and the system administrator's password to the resource manager. **Important:** When changing these passwords, complete the following steps in order for Windows:

1. Log on to the system administration client.
2. Expand the **Resource Manager** tree.
3. Click the resource manager that you want to modify and expand its tree.
4. Click **Server Definitions** and click **Properties**. The Server Definition properties window opens.
5. Change the password in the **Password** field.
6. Click **OK**.
7. Right-click the resource manager that you expanded (in step 3) and click **Properties**. The Resource Manager properties window opens.
8. Change the password in the **Password** field.
9. Click **OK**.

Changing the database access passwords

If you need to change the database access passwords, you need to change the operating system password for the database connection and the ICMRM.properties file so that the resource manager can identify the new password. To change the operating system password on Windows, complete the following steps:

1. Click **Start**→**Settings**→**Control Panel**.
2. Open **Users and Passwords**.
3. Click **ICMRM**.
4. Click **Set Password**.
5. Enter the new password.

To change the ICMRM.properties file, complete the following steps:

1. Open the ICMRM.properties file. The default location is X:\WebSphere\AppServer\installedApps\icrm.ear\icrm.war\WEB-INF\classes\com\ibm\mm\icrm\icrm\ICMRM.properties., where x is the location of the drive in which you installed Content Manager.
2. Change the **DBPassword** to match the operating system password.
3. Save the ICMRM.properties file.

After you change the database password, the database either needs to be restarted, or you can let it issue two or three errors until it resets itself.

Defining language codes

Content Manager requires you to specify a language code if you plan on translating text from one language to another. A language code is a 3-character code that can be used to display attributes or item types in multiple national languages. When you specify a language code, you also need to enter the equivalent word in that language.

The system administration client has several **Display name** fields that have the **Translate** button next to them, one of which is located on the Attribute window. If you have to set up your Content Manager for users who speak different languages, then you need to define these languages to the library server using the table below.

After you define the language codes that your Content Manager system recognizes, you use the **Translate** button to put in the translated terms, changing how that term is viewed by the end-user of the client application. For example, if you have an attribute that you named Street and have Spanish as one of the languages defined to your library server, then you can click the **Translate** button and type

Calle. So, when end-users who use a Spanish version of the client application need to give a value for the attribute *Street*, they will see *Calle* for the attribute instead.

A language code must be one of the 3-character codes shown in Table 19:

Table 19. Language codes available in Content Manager

Language code	Language
AFR	Afrikaans
SQI	Albanian
ARA	Arabic
ENA	Australian English
BEL	Bulgarian
BGR	Byelorussian
CAT	Catalan
CHS	Chinese, Simplified
CHT	Chinese, Traditional
HRV	Croatian
CSY	Czech
CZE	Czech Republic
DAN	Danish
NLD	Dutch
NLB	Dutch, Belgian
ENG	English, United Kingdom
ENU	English, US
ENP	English, uppercase
FIN	Finnish
FRA	French
FRB	French, Belgian
FRC	French, Canadian
FRS	French, Swiss
DEU	German
DES	German, Swiss
ELL	Greek
HEB	Hebrew
HUN	Hungarian
GAE	Irish Gaelic
ISL	Icelandic
ITA	Italian
ITS	Italian, Swiss
JPN	Japanese
KOR	Korean
MKD	Macedonian
NOR	Norwegian Bokmal

Table 19. Language codes available in Content Manager (continued)

Language code	Language
NON	Norwegian Nynorsk
PLK	Polish
PTG	Portuguese
PTB	Portuguese, Brazilian
RMS	Rhaeto-Romanic
ROM	Romanian
RUS	Russian
SRB	Serbian, Cyrillic
SRL	Serbian, Latin
SKY	Slovakian
SLO	Slovenian
ESP	Spanish
SVE	Swedish
THA	Thai
TRK	Turkish
UKR	Ukrainian
URD	Urdu

You must define an attribute in every language that the attribute is used on your system. If an attribute displays in a language that is different from the language defined on a machine, an asterisk (*) displays before the attribute name.

Defining a resource manager

The resource manager is the repository for content stored in the Content Manager system. Users store and retrieve objects to and from the resource manager by issuing requests through the library server. When a request is granted, the library server returns a security token and the location of the object to the users.

When retrieving content, the client uses the security token to access the resource manager and gives the location of the object to find the object. The object is then returned to the client and copied to the staging area.

Furthermore, if a resource manager does not have the object that the client is looking for, then the initial resource manager forwards the request to any other server that it recognizes. After the requested object is found, the object is copied to the staging area of the initial resource manager and sent to the requesting client.

When you define a resource manager or define a server to a resource manager, you need to know the new server's:

- Server name
- Server type
- Host name
- User name
- Access to the server (such as a valid user ID and password)

- Protocol
- Port
- Schema
- Path

Important: The user ID and password of the resource manager that you want to access must match the user ID and password that you used to log onto the system administration client. If the user IDs and passwords are different, then you will get a prompt asking you for the user ID and password for that resource manager. You cannot configure or modify a resource manager unless you have access to it.

To define a specific resource manager, see “Adding a resource manager to the library server” on page 54.

Configuring a resource manager

When you add a resource manager to your library server, you must also configure it. When you configure the resource manager, you define the rules under which it will operate. You define database connections, time outs, cycles of resource manager-related processes such as purger, migrator, asynchronous recovery, and schedule information for migration.

Configuration takes some planning. You must analyze what types of items the resource manager manages and the pattern in which users access these items. Based on your analysis, you can decide when to purge or migrate items. You can set schedules one way now, but as your needs change, you might decide to change your schedules and cycles.

To configure your resource manager, go to the resource manager that you want to configure and select **Configurations**. Right-click **Configurations** and select **New**. The New Resource Manager Configuration window opens. Within this window, you need to specify a configuration file that you use with your resource manager. IBMCONFIG is the default configuration file.

See Chapter 8, “Managing databases”, on page 103 for an overview of scheduling times for purging and migrating items. See the system administration client online help for specific steps to set up purging and migrating your items.

Configuring Secure Sockets Layer

The resource manager requires Secure Sockets Layer for administration. You also need to enable both HTTP and HTTPS access for the resource manager to be fully functional.

You must complete three tasks to configure Secure Sockets Layer. The first of these, creating a key database, is defined below:

1. Enter `ikeyman` on a command line on UNIX[®] or start the Key Management utility in the **IBM HTTP Server** folder on Windows NT.
2. Select **Key Database File** from the main window, then select **New**.
3. Ensure that the directory `c:\key` exists. Then, in the New window, enter your key database name, such as `C:\keys\key.kdb` or click `key.kdb` if you are using the default.
4. Click **OK**.
5. In the Password Prompt window, enter your correct password and press enter.

6. Click **OK**.

The second task in configuring Secure Sockets Layer is to create a self-signed certificate as follows:

1. Enter `keyman` on a command line on UNIX or start the Key Management utility in the **IBM HTTP Server** folder on Windows NT.
2. Click **Key Database File** from the main window, then click **Open**.
3. In the Open window, enter your key database name or click on `key.kdb` to use the default. Click **OK**.
4. In the Password Prompt window, enter your correct password and click **OK**.
5. Click **Personal Certificates** under the Key Database and click **New Self-Signed**.
6. In the Create New Self-Signed Certificate window, enter:
 - **Key label:** enter a descriptive comment used to identify the key and certificate in the database.
 - **Key size.**
 - **Common name:** enter the fully qualified host name of the Web server as the common name. Example: `www.myserver.com`
7. Click **OK**.

The third task, setting up Secure Sockets Layer with the IBM administration server, requires the following steps:

1. Open up the IBM HTTP admin console in a browser window on the HTTP server machine (the default URL is `http://localhost:8008/admin`).
2. Set up a security module with the following steps:
 - a. Click **Basic Settings**.
 - b. Click **Module Sequence** (scope: global).
 - c. Click **Add**.
 - d. From the **Select a module to add** list, select `ibm_ssl`. The module dll is placed to the right.
 - e. Click **Apply**.
 - f. Click **Close**.
 - g. Click **Submit**.
3. Set up a **secure host** IP and additional port for your secure server with the following steps:
 - a. Click **Basic Settings**.
 - b. Click **Advanced Properties** (scope: global).
 - c. Click **Add** for the **Specify additional ports and IP addresses field**. Leave the IP address field empty and enter 443 in the **Port** field.
 - d. Click **Apply**.
 - e. Click **Close**.
 - f. Click **Submit**.
4. Set up a **virtual host** for the secure server:
 - a. Click **Configuration Structure**.
 - b. Click **Create Scope** (scope: global).
 - c. Click **VirtualHost** in the **Select a valid scope to insert within the scope selected in the right panel field**.
 - d. Enter the virtual host IP address or fully qualified domain name.
 - e. Enter the virtual host port (**443**).

- f. Enter the server name.
 - g. Leave alternate name(s) for host blank.
 - h. Click **Submit**.
5. Set up a **virtual host document root** for secure server through the following steps:
 - a. Click **Basic Settings**.
 - b. Click **Core Settings** (scope: virtual host you are working with).
 - c. Enter the server name as a fully qualified domain name.
 - d. Enter the document root directory name.
 - e. Click **Submit**.
 6. Set **keyfile** and **SSL timeout values** for secure server through the following steps:
 - a. Click **Security**.
 - b. Click **Server Security** (scope: global and virtual host).
 - c. Click **No** for **Enable SSL**. This disables SSL for global scope.
 - d. Enter the path and keyfile filename.
 - e. Enter a timeout value for SSL Version 2 session IDs (**100 seconds**).
 - f. Enter a timeout value for SSL Version 3 session IDs (**1000 seconds**).
 - g. Click **Submit**.
 7. Enable SSL and select mode of client authorization through the following steps:
 - a. Click **Security**.
 - b. Click **Host Authorization** (scope: virtual host) host IP addr: 443.
 - c. Click **Yes** for **Enable SSL**. This enables SSL for virtual secure host.
 - d. Click **none** for **Mode of client authorization to be used**.
 - e. Click **Add** on the **Cipher specification(s) that can be used in a secure transaction**. Add specifications 39, 3A, 62, 64.
 - f. Click **Submit**.

After configuring the Secure Sockets Layer, you should test the server installation. To do this, test the HTTP connection by entering `http://your_host/icrm/snoop` to see the snoop information returned. Also, test the HTTPS (SSL) connection by entering `https://your_host/icrm/snoop` to see the snoop information returned here as well.

Cataloging objects from your local machine

Cataloging allows you to store resource manager objects on your local machine. By using the catalog API, you can instruct the resource manager to turn a directory on your machine into another accessible volume.

To catalog, you must complete the following steps:

1. Enable the IBM Catalog Device Manager:
 - a. Right-click **Device Managers**.
 - b. Click **New**.
 - c. Type `ICMFILEPATH` in the Name field.
 - d. Click **Enable**.
 - e. Click **OK**.

2. Create a storage class for cataloging (for example, CATCLASS), specifying ICMFILEPATH as the device manager.
3. Create a migration policy for cataloging (for example, CATMGT). Add your storage class to it (for example, CATCLASS).
4. Create a storage system.
5. Create a storage group.
6. Create a collection for cataloging (for example, CATCOL), specifying your migration policy (for example, CATMGT).
7. Write a program that creates an item and catalogs it. Example:

- a. Create a text resource item type (Journal) with attributes (Title,Year).

```
DKItemTypeDefICM textItemType = new DKItemTypeDefICM(datastore);
textItemType.setName("Journal");
textItemType.setClassification
(DKConstantICM.DK_ICM_ITEMTYPE_CLASS_RESOURCE_ITEM);
textItemType.setXDOClassId(DKConstantICM.DK_ICM_XDO_TEXT_CLASS_ID);
```

```
//add attrs to the item type.
textItemType.addAttr(TitleAttrObj);
textItemType.addAttr(YearAttrObj);
textItemType.add();
```

- b. Create a resource item and catalog content. To catalog file ReadMe.txt with file path c:\winnt existing on the resource manager, enter:

```
DKLobICM lobj = datastore.createDDO("Journal",DKConstant.DK_CM_ITEM);
lobj.catalogContent("ReadMe.txt","c:\winnt");
```

Configuring a media server

Content Manager can manage multimedia objects like scanned documents, images, text, and presentation files. Content Manager can also manage audio and video files (called media objects in Content Manager and assets in VideoCharger) by integrating with VideoCharger. So, Content Manager stores media objects in the VideoCharger Server as assets.

In Content Manager, the VideoCharger Server can bond with the resource manager as a Media Server or Media Resource Manager. To add and configure a VideoCharger Server to Content Manager, see *IBM Content Manager VideoCharger for Multiplatforms Planning and Installing VideoCharger*.

Staging area

You use the staging area as a local area network (LAN) cache and a place to retrieve objects stored on Tivoli® Storage Manager (TSM). Staging areas need fast disk drives for high-demand objects, large objects, and objects that require high-speed performance to access, like audio and video objects. Staging areas provide fast performance and allow you to access large objects that could be stored on slower devices.

The staging area is created when you install Content Manager. The system administration client allows you to configure the staging area for size and purge rate. You can only have one staging area for each resource manager.

A client application requests an item directly from a resource manager. If the resource manager does not have the item located on its storage system, it requests the location from the library server. The library server, having stored the item type metadata, will know the location of the item on another resource manager and

provide the location to the requesting resource manager. The requesting resource manager retrieves the item (assuming that it has access to that resource manager) from the location and places it in its staging area. When the request for the item comes again, then, for faster retrieval, it can retrieve it from a local cache and return it to the client.

Chapter 4. Managing object storage

Content Manager allows you to store multiple copies of items (objects) and migrate items from one storage location to another. You plan which items to replicate or migrate at the time that you store the item.

When you manage object storage, you create the collections that organize the items in your system and you create the migration policies that move those items from one type of storage to another. A *collection* identifies a group of items.

Other tasks included in managing object storage are determining which media to use to store the items and identifying the schedule for moving the items from one media type to another.

Figure 29 shows you the flow of a store request. The library server logs the request and moves the request and the item to the resource manager. The resource manager then logs the location of the item and sends it to the storage subsystem for storing.

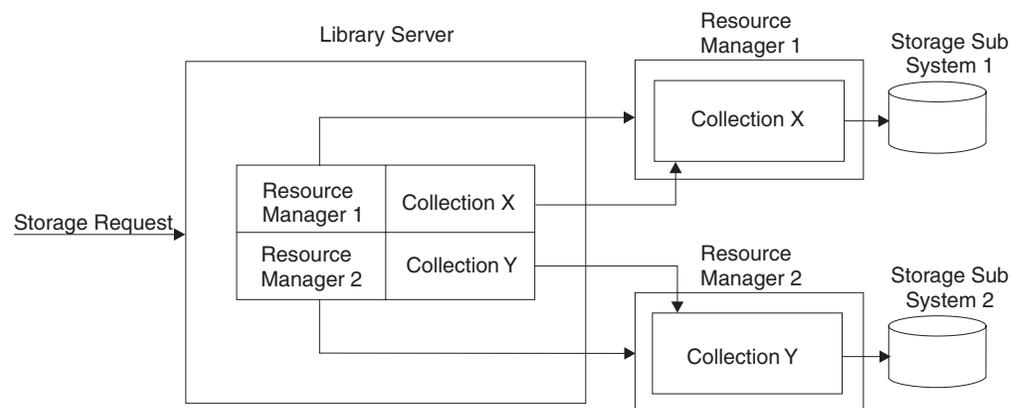


Figure 29. When you store an item, the library server and resource manager log where the item is located.

You migrate items from high-speed storage devices to low-speed storage devices because storing all items on high-speed devices is expensive. You need to reserve the high-speed storage devices for those items that users need to use frequently and, in the case of large media objects, for those items that need the high-speed performance to play videos or return frequently requested large objects quickly. Large items and less frequently used items reside on the slower, yet larger, storage devices.

You must also consider the length of time that you want to keep your content. For example, microfiche can reportedly last 500 years, while content on hard disks degrades much faster.

Device managers

A device manager is the interface between the resource manager and the storage system. It communicates the tasks that you define for the resource manager to the storage system where you store your objects. The dynamic link library (DLL) or shared library for the device manager must be installed on the workstation where the resource manager is installed before the device manager is created in Content Manager.

Table 20 shows the possible device managers and the operating systems on which you can use them. Content Manager installs all the device manager listed in the table; however, most of the device managers are disabled. ICMHDDM and ICMFILEPATH install as enabled device managers whereas ICMADM, ICMVCDM, and ICMADDM install as disabled.

Table 20. Device managers and the operating systems they work on

Device manager	Operating system
ICMHDDM	Windows NT
GPFS	AIX 5
JFS	AIX
ICMMADM	Media Archiver
OAM	OS/390®
ICMADDM	Tivoli Storage Manager
ICMVCDM	VideoCharger
ICMFILEPATH	Catalog
ICMREMOTE	Remote server

If the device manager is disabled, the storage systems that use that device manager are inaccessible to the resource manager. You cannot store any new objects on the storage system, and you cannot retrieve any existing objects. You might want to disable a device manager in the following situations:

- When you do not have the specific device manager installed
- When you do not have the specific storage system available
- When you want to perform maintenance and you do not want users to access the storage systems that are associated with the device manager

Use the system administration client to create the device managers that you need to access your storage systems. You can assign a device manager to as many storage classes as you want, but a storage class can only have one device manager.

Storage classes

A storage class identifies the type of media that an object is stored on. It is not directly associated with a physical location; however, it is directly associated with the device manager, which is the interface between the resource manager and the actual physical location. Types of storage classes include:

- Fixed disk (DASD)
- Optical
- Stream
- Tape

- Tivoli Storage Manager (TSM)

A *storage system*, which is grouped together with a storage class by a storage group, represents an actual physical device or unit in which the objects in our system are stored.

When you decide to migrate objects from one storage system to another, you can move them locally or remotely. When you move them locally, the Content Manager system provides a list of installed device managers that you can choose to associate with the storage class.

When you choose a remote location to create a storage class, you must know the resource manager and collection in which you want objects to move. You cannot assign a device manager to a remote storage class because the device managers are unique to the resource managers in which they are installed. You need to create a valid storage class on the remote resource manager to handle the objects that you want to migrate.

You must associate a storage class to a storage group. For more information, see “Storage groups” on page 66.

Use the system administration client to create the storage classes for the types of media that you have. You can assign one and only one device manager to each storage class.

Storage systems

A *storage system*, also known as a volume, is the location where an object is stored. For example, on Windows, a storage system is defined as a physical or logical partition on a hard disk drive.

Objects need to exist on certain types of storage systems to retain their integrity. For this reason, Content Manager has four storage systems to which you can store objects:

Table 21. Select the storage systems for use with Content Manager

	AIX	Windows	VideoCharger
File system	X	X	
Media archive	X	X	X
Tivoli Storage Manager (TSM)	X	X	
VideoCharger			X

You must associate a storage system to a storage group. For more information about storage groups, see “Storage groups” on page 66.

Storage systems have four different assignments: unassigned, overflow, assigned, and offline. Unassigned identifies a space on a system, but does not assign it to a storage group. In this case, the resource manager cannot recognize the storage system. This assignment is useful if you want to define several storage systems that you do not have yet or if you do not want to use them at the time that you create them.

When you assign overflow to a storage system, you indicate that the storage system is available to a storage group but it does not have space enough to hold the objects that it is receiving.

When you assign a storage system to a storage group, the storage system belongs to that group. You can assign a storage system to one or more storage groups.

When you assign a storage system as offline, you indicate that the storage system is not mounted or is temporarily unavailable. For example, if you had a disk drive that you could remove from a machine, then you could indicate the disk drive as "offline" when you detach it so that users cannot store or retrieve from it. Or, if the LAN connection to a storage system is down, you might have to temporarily take that storage system offline.

Use the system administration client to create the storage systems to store your objects. You must have a storage class already defined when you create your storage system. See the system administration client online help to help you define storage classes.

Storage groups

Storage groups contain the identities of the storage systems and storage classes that you use to store the objects in a collection. A storage group is one of two essential components that creates a collection. The other component that creates a collection is the migration policy. The migration policy is the path that objects take when they move from one storage class to another. For example, you could have storage groups for high demand data and storage groups in low demand data (DASD vs. tape).

A storage group contains one or more storage systems and storage classes. It associates each storage system to a storage class. The migration policy contains a list of storage classes. Through the storage class to storage system association, the objects know the storage system to which they belong, and through the migration policy, they know the storage system to which they will move next.

Use the system administration client to create the storage groups that you need. You must have a storage system and a storage class defined to create a storage group. See the system administration client online help to help you define storage classes and storage systems. **Recommendation:** When you create your Content Manager system, assign a different storage system for each storage group and a different storage group for each collection.

Migration policies

A migration policy contains the rules for migrating the objects in a collection. It requires one or more storage classes, which you must create first. Tivoli Storage Management (TSM) calls its migration policies management classes.

The migration policy defines how long an object stays in a location and where the object will move next. The storage class determines the location. The location is limited to the storage systems in the storage group that is assigned to the collection to which the object belongs.

To migrate an object to another resource manager, specify a remote storage class as the final step in a migration policy. See "Setting up remote migration" on page 106 for more information.

Use the system administration client to create the migration policies that you need. You can use the same migration policy for more than one collection. You must have a storage class defined to create a migration policy. See the system administration client online help to define storage classes.

Collections

The collection is the last component that you define for object storage because it requires a storage group and migration policy, which you must create first.

A collection identifies a group of related objects with similar storage management criteria. All objects in a collection are stored on the storage systems specified in the storage group of that collection. All objects in the collection migrate according to the rules that are defined for the migration policy in that collection.

You must have a migration policy and a storage group defined to create a collection. See the system administration client online help to define migration policies and storage groups.

Use the system administration client to create the collections that you need to logically group the objects in your system.

Replication

For enhanced retrievability and security, 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. More specific information about defining these settings is provided in the online help.

When you define a resource manager in the New Resource Manager Definition window of the system administration client, you can mark a resource manager as unavailable. You might want to do this if the server crashed or is under maintenance. If you do this, a client bypasses this server and does not store or retrieve objects in it. Also, in the New Library Server Configuration window, you can set the number of seconds that the library server waits to check for the availability of resource managers and number of seconds that it waits for a response from the resource manager before considering it to be unavailable.

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. **Recommendation:** Run the replicator during times when there is little server activity.

Creating server definitions

In order for replication to work, you must define your resource managers to the library server, define each resource manager to each other, and then define your

collections. **Example:** Your primary resource manager is RMDB1. Your two replica resource managers are Rep1 and Rep2. To create these definitions:

1. Define your resource manager to the library server by opening the New Resource Manager Definition window in the system administration client. Type RMDB1 in the **Name** field and complete the rest of the fields such as the **Hostname** field to connect to the resource manager and the **User ID** field to logon to it. **Attention:** See the online help for specific help. **Restriction:** Each of the primary and replica resource managers must point to the same library server.
2. Repeat the process used in Step 1 to define Rep1 and Rep2 to the library server.
3. Expand the tree node in the System Administration Client for RMDB1. Right-click the Server Definition node for RMDB1, open the New Server Definition window, and add the information (Name, Server Type, Hostname, User ID, Password, and so forth) for Rep 1 so that RMDB1 can communicate with it. You are adding the Rep1 server information to RMDB1.
4. Open a second New Server Definition window and add the Rep2 server information to RMDB1.
5. Expand the tree nodes for Rep1 and Rep2 and repeat the process used in Steps 3 and 4 to create server definitions for Rep1 and Rep2. You do this so that these replica resource managers know about RMDB1 and each other.
6. Expand the RMDB1 tree node, right-click Workstation Collections to open the New Workstation Collection window to create a collection for RMDB1.
7. Expand the Rep1 and Rep2 tree nodes and repeat the process used in Step 6 to create collections for Rep1 and Rep2.
8. Click **Add** from the RMDB1 Workstation Collection Properties window to open the New Workstation Collection Entry window. Here you enter the target resource manager to which you want to replicate and the target collection in the target resource manager (such as Rep1, Collection 1). For example, you can replicate object data in RMDB1, Collection 1 to Rep 1 Collection 1. You can also replicate object data in RMDB1, Collection 1 to Rep1, Collection 2, and so forth.

Library server monitor fail-over service

Content Manager provides a fail-over service that verifies that resource managers are available. If you are trying to store objects into a resource manager that is unavailable, then 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). To start the service:

- On Windows, You can check to see that it is started from the Services panel.
- On AIX, make sure that `icmx1sap` is running.
- On Solaris, make sure that `icms1sap` is running.

Turning on replication for objects that have already been stored

This process is should be only attempted after backing up your systems. You should replicate small batches of objects off the same media to ensure maximum

efficiency. At first, you should use this procedure when you are the only system user so that you can monitor the replication rate and determine how many objects you can replicate at a time.

If you are going to enable collections for replication, replicate to the same server, or cross replicate between servers that have collections that contain both primary and replicated parts, you should make a copy of your current robjects tables. This copy then be used to distinguish between primary and replicated parts.

Restrictions:

- This procedure only works for primary parts. It is not possible to tell only from resource manager data if a part is a primary or replica part. You must be able to use some group of attributes to determine which parts that you have replicated and which parts are primary parts that have yet to be replicated.
Recommendation: Make target collections accept replicated data and keep that replicated data separate from primary copies.
- Determine which parts to replicate and the target resource manager and target collection on the target server.
- Allow for storage space on the target server.
- Ensure the space exists for the DB2 tables and logs.
- For remote migration, have entries for the remote resource managers.
- **Important:** Do not allow discards of the objects that are being replicated via this process until replication is complete. Otherwise you might have requests to replicate objects that do not exist. This can result in replication being unable to process these records. If this happens, the records need to be identified and removed by hand from the rmreplication table.

To manually enable existing objects for replication:

1. Run the migrator. If you have objects of status S,U,or D, the migrator has not completed its work. Do not attempt to replicate.
2. Run the replicator twice. The base_replication table should be empty.
3. Backup the entire system including the library server and both source and target resource managers.
4. On the source resource manager, using a DB2 command line or SQL Plus command, connect to the source resource manager database.
5. Determine the distribution of objects by collection. Run the following query to get a collection/volume distribution:

```
select col_collname, obj_volumeid, count(*) from robjects a,  
base_collections b where a.obj_collectionid = b.col_collid  
and obj_status = 'A'  
group by col_collname, obj_volumeid  
order by col_collname, obj_volumeid
```

6. Run the following query to get a collection/volume/date distribution substituting the 'SOURCE_COLLECTION' name that you wish to replicate:

```
select col_collname, obj_volumeid,DATE(obj_createdate),  
count(*) from robjects a  
base_collections b where a.obj_collectionid = b.col_collid  
and obj_status = 'A' and b.col_collname = 'SOURCE_COLLECTION'  
group by col_collname, obj_volumeid ,DATE(obj_createdate)  
order by col_collname, obj_volumeid ,DATE(obj_createdate)
```

Choose a collection, volume, and date range to replicate. For the first time, keep the number small. You might increase the numbers once you are certain things are setup and working correctly.

7. Run the insert that places the requests that replicates the chosen objects.
 - replace 'TARGETRM' with your target resource manager db name (upper case)
 - replace 'TARGET.COLL' with your target OS collection (upper case)
 - replace 1 with the volume that you have selected
 - replace the timestamp values with the data range that you have selected

Note: For Oracle, you need to use an Oracle compatible date.

```
insert into rmreplication select obj_libraryid,
obj_itemid, obj_version, obj_collectionid,
'TARGET.COLL' , b.svr_serverid , 'N' ,
obj_size , obj_updatedate from rmojects a,
rmserver b where
b.svr_servername = 'TARGETRM'
and obj_status = 'A' and obj_volumeid = 1
and obj_createdate between
'2003-01-01-00.00.00.000000' and '2003-01-30-00.00.00.000000';
```

If you make a spelling error you may need to remove the problem rows from base_replication. If you leave rows that cannot be processed, the replicator might not function correctly.

8. Run `select count(*) from base_replication`
9. Run the replicator. The replicator will start by updating the library server. The rmreplications table will then have a REP_REPLICATIONTYPE of 'R'. The objects should begin to store on the target server.
10. Verify that the parts have arrived at the target object server and that the rmreplication table is empty.

Defining replication rules in administrative domains

For a user to enable replication, the source and target resource managers/collections must be in user's own domain or the PUBLIC domain. If the user is in the Super Domain, the user can define a replication rule in any domain, but the source and target must be in the same domain or one of them must be in the PUBLIC domain.

Lan cache

The Content Manager system administration client has a feature that allows users to enable LAN cache. If you have end users who frequently retrieve the same object, enabling LAN cache can improve end-user efficiency by reducing the time required to retrieve and display an object stored on a remote content server.

You can enable LAN cache from the New Resource Manager Definition window in the system administration client. When you enable LAN cache, the Content Manager system retrieves the requested object from the remote server and stores the object in the staging directory of the server that supports the local resource manager. When client users request the object, the system retrieves the local copy, instead of accessing the original image on the remote server.

| Each time a client attempts to retrieve the cached object, the resource manager
| compares the timestamp applied when the object was originally retrieved to the
| timestamp of the object on the remote server. If the timestamps are different, the
| resource manager retrieves the updated object and overwrites the original cached
| object.

| For example, your system has three client users who are working on an insurance
| claim. Each user needs to view the same large photograph of a damaged car. The
| photograph, which is in the .TIFF file format, is stored on a content server in a
| different state.

| If LAN cache is not enabled, each client user requests and receives the file from the
| remote server. Depending on file size and network traffic, the retrieval and display
| process can be slow and could reduce efficiency of the client users. With LAN
| cache enabled, each client user receives a copy of the object stored on the local
| resource manager.

| The system administration client also allows users to manage the staging directory
| to get the most benefits from LAN caching. Staging directory management tasks
| include:

- | • setting automatic cache purge specifications: A purge removes the oldest, least
| frequently used objects from the staging directory.
- | • defining subdirectories to hold cached objects: Storing cached objects in
| subdirectories can improve system retrieval time because the system can target
| the search without looking through individual objects stored in the staging
| directory.
- | • defining the size of the staging directory: Depending on the size and volume of
| cached objects, you might need to modify the original parameters defined for
| the staging directory.
- | • defining the maximum size of the cached object: The system will not cache
| objects that exceed the maximum size. However, if you decrease the maximum
| size and objects that were stored earlier exceed the new maximum size, the
| system will retain the objects.

Chapter 5. Managing servers

You must maintain the quality and integrity of the system. To maintain the system, your responsibilities include:

- Starting and stopping servers
- Synchronizing servers
- Running the asynchronous recovery utility
- Backing up and restoring your data
- Tracing errors
- Replacing or repartitioning a hard disk

Some of these responsibilities require that you work with your database administrator.

Starting and stopping servers

You might find that you have to restart your servers. Reasons to restart your server can include:

- Picking up changes that you made to the WebSphere® configuration file
- Stopping a server from dumping a huge amount of data in an abnormal termination
- Installing a new WAR file
- Changing the `icmm.properties` file

When you decide to restart your server, consider the amount of time it takes to restart it. Consolidate any changes so your servers are down for the least possible span of time.

Starting and stopping a Windows server

You can install a server as a servlet or as a stand-alone application. Each choice has a different method of starting and stopping a server. The following procedure explains how to stop a Windows stand-alone application. Windows NT and Windows 2000 have a few different steps.

You must grant a user logon service access before that user can start or stop a server. To grant this access for Windows NT, follow these steps:

1. Click **Start** → **Programs** → **Administrative Tools** → **User Manager**.
2. Click on **Policies**, then select **User Rights**.
3. Check the box by **Show Advanced User Rights**.
4. Under the **Right** scrolling window, select **Log on as a service**.
5. Click **Add**.
6. Click **Show users**.
7. Select the user that you want to add.
8. Click **Add**.
9. Click **OK**.
10. Click **OK** once more to complete the process.

To grant logon service access on Windows 2000, follow these steps:

1. Click **Start** → **Settings** → **Control Panel**.
2. Double-click **Administrative Tools**.
3. Open **Local Policies**.
4. Open **User Rights Assignment**.
5. Open **Log On as a Service**.
6. Select the name you want to add and click **Add**.

After you have service access, you can start or stop a server on Windows NT through the following process:

1. Click **Start** → **Settings** → **Control Panel**.
2. Double-click **Services**.
3. Select the server that you wish to start or stop.
4. Left-click the appropriate button to start or stop the server.

You can start or stop a server on Windows 2000 through the following process:

1. Move your cursor to **My Computer** and right-click on the icon.
2. Click **Manage**.
3. Double-click **Services and Applications**.
4. Right-click on the server you wish to start or stop, then click the appropriate choice.

You can set the server to automatically start on Windows NT through the following steps:

1. Click **Start** → **Settings** → **Control Panel**.
2. Double-click **Services**.
3. Select the server that you want to configure as automatic or manual.
4. Click **Startup**.
5. Select the radio button appropriate to your choice of automatic, manual or disabled.
6. Click **OK**.

You need to start or stop servlet applications on Windows through WebSphere Version 4.0 Advanced Edition, WebSphere 4.0 Advanced Edition Single Server, or WebSphere 5. Follow these steps to start or stop an application in WebSphere:

1. *For WebSphere Version 4.0 Advanced Edition:* Check that WebSphere Application Server is running. If it is not, start it: **Start** → **Programs** → **IBM WebSphere** → **Application Server V4.0 AE** → **Start Admin Server**.
For WebSphere Version 4.0 Advanced Edition Single Server: Check that WebSphere Application Server is running. If it is not, start it with the following script: `c:\WebSphere\AppServer\bin\startupServer.bat`
For WebSphere Version 5: Check that the WebSphere Application Server is running. If it is not, start it: **Start** → **Programs** → **IBM WebSphere** → **Application Server V5.0 AE** → **Start the Server**.
2. *For WebSphere Version 4;* ensure that `<install_disk>:\WebSphere\AppServer\logs\tracefile` contains the line:
`Server __adminServer open for e-business.`

For WebSphere Version 5:, ensure that `<install_disk>:/Program Files/WebSphere/AppServer/logs/server1/startServer.txt`, where `server1` is the default server name, contains the line:

`Server __adminServer open for e-business.`

3. *For WebSphere Version 4.0 Advanced Edition:* Start the WebSphere Advanced Administrative Console with **Start** → **Programs** → **IBM WebSphere** → **Application Server V4.0AE** → **Administrator's Console**.

For WebSphere Version 4.0 Advanced Edition Single Server: Open the Web page `http://hostname:9090/admin` where `hostname` is the name of your host machine.

4. *For WebSphere Version 4.0 Advanced Edition:* The resource manager servlet Web application is called `icmrm` under **Nodes** → **bc1** → **Application Servers**. Right-click this server, then click either **start** or **stop** to start or stop the server.

For WebSphere Version 4.0 Advanced Edition Single Server: The resource manager servlet Web application is called `icmrm`. Select the check box for this option, then click either **start** or **stop** to start or stop the server.

For WebSphere Version 5: The resource manager servlet Web application is called `icmrm` under **Applications** → **Enterprise Applications**. Select the check box to start or stop the server.

Starting and stopping an AIX server

You can install a server as a servlet or as a stand-alone application. Each choice has a different method of starting and stopping a server. The following procedure explains how to start or stop an AIX servlet application.

1. *For WebSphere Advanced Edition:* Check that WebSphere Application Server is running. If it is not, start it by running the following script:

```
/usr/WebSphere/AppServer/bin/startupServer.sh
```

For WebSphere Advanced Edition Single Server: Check that WebSphere Application Server is running. If it is not, start it by running the following script: `/usr/WebSphere/AppServer/bin/startServer.sh`

2. Ensure that `/usr/WebSphere/AppServer/logs/tracefile` contains the line:

`Server __adminServer open for e-business.`

3. *For WebSphere Advanced Edition:* Start the WebSphere Advanced Administrative Console with `/usr/WebSphere/AppServer/bin/adminclient.sh`.

For WebSphere Advanced Edition Single Server: Open the Web page `http://hostname:9090/admin` where `hostname` is the name of your host machine.

4. *For WebSphere Advanced Edition:* The resource manager servlet Web application is called `icmrm` under **Nodes** → **bc1** → **Application Servers**. Right-click this server, then click either **start** or **stop** to start or stop the server.

For WebSphere Advanced Edition Single Server: The resource manager servlet Web application is called `icmrm`. Select the check box for this option, then click either **start** or **stop** to start or stop the server.

There are four stand-alone applications: `RMMigrator`, `RMPurger`, `RMReplicator`, and `RMStager`. The following procedure explains how to start or stop an AIX stand-alone application.

1. There is a procedure to start or stop all four applications at once on any of the resource manager databases.

- a. To **start** all four applications, enter the following command:

```
/etc/rc.cmrproc start dbname rmwebpath
```

This will start all four applications on `dbname` and `rmwebpath`.

- b. To **stop** all four applications, enter the following command:

```
/etc/rc.cmrmproc stop dbname rmwebpath
```

This will stop all four applications on *dbname* and *rmwebpath*.

2. This procedure will allow you to start or stop applications selectively.

- a. To **start** an application, enter the following command:

```
/etc/rc.cmrmproc start dbname rmwebpath application
```

where *dbname* is the database name on which these processes are running; *rmwebpath* is the context root that was selected when installing Content Manager; *application* is the resource manager stand-alone application that you want to start.

For example, `/etc/rc.cmrmproc start rmdb icmrm RMMigrator` starts the resource manager migrator on the database *rmdb* with the *rmwebpath icmrm*.

- b. To **stop** an application, enter the following command:

```
/etc/rc.cmrmproc stop dbname rmwebpath application
```

where *dbname* is the database name on which these processes are running; *rmwebpath* is the context root that was selected when installing Content Manager; *application* is the resource manager stand-alone application that you want to stop.

For example, `/etc/rc.cmrmproc stop rmdb icmrm RMMigrator` stops the resource manager migrator on the database *rmdb* with the *rmwebpath icmrm*.

Starting and stopping a server on the Solaris Operating Environment

You can install a server as a servlet or as a stand-alone application. Each choice has a different method of starting and stopping a server. The following procedure explains how to start or stop a Solaris servlet application.

1. *For WebSphere Advanced Edition:* Check that WebSphere Application Server is running. If it is not, start it by running the following script:

```
/opt/WebSphere/AppServer/bin/startupServer.sh
```

For WebSphere Advanced Edition Single Server: Check that WebSphere Application Server is running. If it is not, start it by running the following script: `/opt/WebSphere/AppServer/bin/startServer.sh`

2. Ensure that the `/opt/WebSphere/AppServer/logs/tracefile` contains the line:

```
Server __adminServer open for e-business.
```

3. *For WebSphere Advanced Edition:* Start the WebSphere Advanced Administrative Console with `/opt/WebSphere/AppServer/bin/adminclient.sh`.

For WebSphere Advanced Edition Single Server: Open the Web page `http://hostname:9090/admin`

where *hostname* is the name of your host machine.

4. *For WebSphere Advanced Edition:* The resource manager servlet Web application is called *icmrm* under **Nodes** → **bc1** → **Application Servers**. Right-click this server, then click either **start** or **stop** to start or stop the server.

For WebSphere Advanced Edition Single Server: The resource manager servlet Web application is called *icmrm*. Select the check box for this option, then click either **start** or **stop** to start or stop the server.

There are four stand-alone applications: RMMigrator, RMPurger, RMReplicator, and RMStager. The following procedure explains how to start or stop an AIX stand-alone application.

1. There is a procedure to start or stop all four applications at once on any of the resource manager databases.

- a. To **start** all four applications, enter the following command:

```
/etc/rc.cmrmproc start dbname rmwebpath
```

This will start all four applications on *dbname* and *rmwebpath*.

- b. To **stop** all four applications, enter the following command:

```
/etc/rc.cmrmproc stop dbname rmwebpath
```

This will stop all four applications on *dbname* and *rmwebpath*.

2. This procedure will allow you to start or stop applications selectively.

- a. To **start** an application, enter the following command:

```
/etc/rc.cmrmproc start dbname rmwebpath application
```

where *dbname* is the database name on which these processes are running; *rmwebpath* is the context root that was selected when installing Content Manager; *application* is the resource manager stand-alone application that you want to start.

For example, `/etc/rc.cmrmproc start rmdb icmrm RMMigrator` starts the resource manager migrator on the database `rmdb` with the `rmwebpath icmrm`.

- b. To **stop** an application, enter the following command:

```
/etc/rc.cmrmproc stop dbname rmwebpath application
```

where *dbname* is the database name on which these processes are running; *rmwebpath* is the context root that was selected when installing Content Manager; *application* is the resource manager stand-alone application that you want to stop.

For example, `/etc/rc.cmrmproc stop rmdb icmrm RMMigrator` stops the resource manager migrator on the database `rmdb` with the `rmwebpath icmrm`.

Synchronizing servers

Periodically, you should check that the resource manager and the library server contain consistent information. See “Asynchronous Recovery utility overview” on page 83 for more information. **Recommendation:** In a production environment, synchronize the servers prior to any system backup.

After running the asynchronous recovery program, run the RUNSTATS function for your database. You run RUNSTATS to ensure that your system runs efficiently. Contact your database administrator to assist you on running RUNSTATS.

Backing up and restoring your data

Because everyone has different hardware, media, and policies for backing up data, Content Manager provides a prototype of backup and restore procedures so that you can develop the solution that you need.

To back up and restore data on the resource manager, you can use the Tivoli Storage Manager (TSM) or any other archive utility that is available on your system.

The shared libraries that were created during the running of the library server should also be backed up. The shared libraries are located in %ICMDLL%/database-name/DLL on AIX and in %ICMR00T%\database-name\DLL on Windows NT, where *database-name* is the name of the library server database.

To back up the databases for the servers, use the utilities that are provided with your database software.

Tracing errors

You can turn on two logs, the trace log and the event log. To turn on the event log, select the **Allow system administrator event logging** check box on the Log and Trace page of the Library Server Configuration window. Library server events are logged into the ICMSTITEMEVENTS table. To turn on the trace log, select at least one of the check boxes on the page. The trace information is logged in the file displayed in the **Trace file name** field. You can select a different file name.

The administrator enables tracing and sets the maximum level allowed. Tracing is done only when requested by client applications. It is also possible to directly update the system control table to trace all connections. For more information, contact IBM Support.

Replacing or repartitioning a hard disk

If a volume or file system that is used by your resource manager becomes full, you can replace or repartition the physical disk on which it is located to make more space available.

Replacing or repartitioning the disk makes the information stored in the volumes table (RMVOLUMES) for that volume or file system invalid. When updating the resource manager volumes, do not run the destager at any point of this process. Otherwise, the volumes will not be the same. Use the following procedures to update the information in the volumes table.

For AIX/Solaris:

Replacing the staging volume: The directory for the STAGING volume is in the resource manager database table, *rmstaging*. Follow these steps to replace the staging volumes:

1. Change permissions on /az/vi/staging directory to match those of your resource manager ID or what is currently in place for /home/icmadmin/ubosstg.
2. If all files in /home/icmadmin/lbosstg are currently R/W, you can skip this step because these files have been destaged already. Otherwise, copy all existing files to the new staging volume: `cp -p /home/icmadmin/staging /az/vi/staging`
3. Update the location of your staging volume in the resource manager ID Database:

```
db2 "connect to rmdb user icmadmin using password"
db2 "update rmstaging set sta_path='/az/vi/staging/'"
```

Replacing the storage volume: The resource manager uses the `vol_path` + the `string_table` value of `lbosdata` + `collection` + `num_bucket_value` to develop the path. The `logical_volume` and `mount_point` are used in various calls to get file system information. Follow these steps to update the resource manager storage volume:

1. Change permissions on `/az/vi/data1` directory to match those of your resource manager ID or what is currently in place for `/home/icmadmin/lbosdata`.

2. Copy all existing files to the new storage volume:

```
cp -rp /home/icmadmin/lbosdata /az/vi/data1
```

3. Update the location of your storage volume in the resource manager database. Use `df -k` to determine the `FILESYSTEM` and `MOUNTED ON` location for `/az/vi/data1`. To update the storage volume, follow this example using your files in place of the example one (`/dev/az/dat1`):

```
db2 "connect to rmdb user icmadmin using password"
```

where `icmadmin` is the user ID used to connect to the database and `password` is the password for the user ID.

```
db2 "select vol_volumeid,vol_logicalname,vol_mountpoint from rmvolumes"
```

4. Determine which `VOLUMEID` is the one you need to change. For example, to change `VOLUMEID=1`, enter:

```
db2 "update rmvolumes set vol_logicalname='/dev/az/data1' where vol_volumeid=1"
```

```
db2 "update rmvolumes set vol_mountpoint='/az/vi/data1' where vol_volumeid=1"
```

```
db2 "update rmvolumes set vol_size=0 where vol_volumeid=1"
```

```
db2 "update rmvolumes set vol_path='/az/vi/data1' where vol_volumeid=1"
```

```
db2 "update rmvolumes set vol_freespace=0 where vol_volumeid=1"
```

Notice that the latter two steps force the resource manager to recalculate the volume space and capacity during any new stores. These values are reflected in the `RMVOLUMES` tables when the resource manager shuts down.

For Windows:

Replacing the staging volume: The directory for the `STAGING` volume is in the resource manager database table (`rmstaging`). Follow these steps to replace the staging volumes:

1. Change permissions on `e:\staging` directory to match those of your resource manager ID or what is currently in place for `d:\staging`.

2. If all files in `d:\staging` are currently `R/W`, you can skip this step since these files have been destaged already. Otherwise, copy all existing files to the new staging volume:

```
copy -p d:\staging e:\staging
```

3. Update the location of your staging volume in the resource manager database:

```
db2 "connect to rmdb user icmadmin using password"
```

```
db2 "update rmstaging set sta_path=e:\staging"
```

Replacing the storage volume: If you replace or repartition the hard disk that contains the `LBOSDATA` directory, you need to identify the new configuration to your system:

1. Restore the `LBOSDATA` directory to the new disk or partition.

2. Manually edit the volumes table to change the following columns to zero for the volume that has been changed:

```
VOL_SIZE= 0
```

```
VOL_FREESPACE = 0
```

3. The next time the resource manager writes or deletes an object, the information is read from the new disk or partition and placed in the volumes table.

If your volume is on a different partition, then manually edit the RMVOLUMES table to update the VOL_LOGICALNAME and VOL_MOUNTPOINT.

For example, assume the volume you wish to replace is defined in the RMVOLUMES table entry with VOL_VOLUMEID=1. Then if your new partition is F and this partition is labeled FDRIVE, enter:

```
UPDATE RMVOLUMES set VOL_LOGICALNAME='FDRIVE' where vol_volumeid=1"
```

```
UPDATE RMVOLUMES set VOL_MOUNTPOINT='f:' where vol_volumeid=1"
```

4. Start the resource manager.

Chapter 6. Managing resource manager utilities and services

This section describes a number of utilities and processes that are installed on the Content Manager resource manager. The utilities are available on Windows, AIX, and Solaris. Some of the utilities exist as services on Windows. For all of the other utilities, you must log on to the server where the resource manager is installed. You must log on with a user ID that has DB2 admin privileges.

The utilities and processes included:

- the migrator, purger, replicator, and stager
- Asynchronous Recovery utilities
- Resource manager/library server (RM/LS) utility and the resource manager volume (RM/V) validation utility. These two utilities are installed with the Content Manager resource manager.

General configuration of resource manager utilities and services

This section provides general background about configuring resource manager utilities and services on AIX, Solaris, and Windows.

Configuration for AIX and Solaris

All of these stand-alone application services, the Asynchronous Recovery Utilities, and the Validation Utilities depend on one central file for environment setup. This file is `setprocenv.sh` located in the `$ICMROOT/bin` dir.

Users should ensure that the variables in this file are set to the right values to reflect their environment. Here is a list of variables in `setprocenv.sh` with a description of each:

rmappname

resource manager application name

dbname

resource manager database name

waittime

time that the application process main thread waits for the child threads to shutdown before terminating itself.

INSTHOME

DB2 Instance home directory, used for the resource manager database

ORA_JDBCPATH

If the resource manager database is an Oracle database, set the fully qualified path for the Oracle JDBC location (Oracle JDBC 9.0.x is the prerequisite).

nodename

If using WebSphere 5.0.x, set the WebSphere nodename.

Configuration for Windows

Only the Asynchronous Recovery utility and the Resource Manager validation utility depend on one central file for environment setup. This file is `setprocenv.bat`, located in the `%ICMROOT%/CONFIG` directory.

Users should ensure that the variables in this file are set to the right values to reflect their environment. Here is a list of variables in `setprocenv.bat` with a description of each:

rmappname

resource manager application name

dbname

resource manager database name

waittime

time that the application process main thread waits for the child threads to shutdown before terminating itself.

DB2_JDBCPATH

Fully qualified path for DB2's JDBC location.

ORA_JDBCPATH

If the resource manager database is an Oracle database, set the fully qualified path for the Oracle JDBC location (Oracle JDBC 9.0.x is the prerequisite).

nodename

If using WebSphere 5.0.x, set the WebSphere nodename.

Resource manager services

There are four stand-alone applications: RMMigrator, RMPurger, RMReplicator, and RMStager.

Configuring the resource manager services on AIX or Solaris

In general, the resource manager processes are configured using the `setprocenv.sh` file described in "General configuration of resource manager utilities and services" on page 81. However, the values for `dbname` and `rmappname` can be changed if passed into the Process starting routine. These parameters will override the ones that are set by the `$ICMROOT/bin/serprocenv.sh` file.

Attention: On AIX, all of the parameters `dbname`, `rmappname` and application are case-sensitive. All of the process service names are registered in the `/etc/services` file. Below is an example of how an entry for the services file appears:

```
RMMigrator_RMDB 7500/tcp #Resource Manager Migrator
```

In the example, RMMigrator is the stand-alone application process and RMDB is the database name. The `dbname` and application params passed to the `/etc/rc.cmrmproc` script should match the case in the service name registration in the `/etc/services` file.

Starting and stopping resource services on AIX or Solaris

You can start or stop a stand-alone application process. To start or stop all four applications at once on any of the resource manager databases:

- Enter the `/etc/rc.cmrmproc start` command to start all four applications using the default values for `dbname` and `rmappname`, specified in the `$ICMROOT/bin/setprocenv.sh` file.
- Enter the `/etc/rc.cmrmproc start dbname rmappname` command to start all four applications on `dbname` and `rmappname`.
- Enter the `/etc/rc.cmrmproc stop dbname rmappname` command to stop all four applications on `dbname` and `rmappname`.

To start all applications selectively, enter the `etc/rc.cmrproc start dbname rmapname application` command, where `dbname` is the database name on which these processes are running; `rmapname` is the name of the resource manager web application; and `application` is the resource manager stand-alone process that you want to start. For example, `/etc/rc.cmrproc start rmdb icmrm RMMigrator` starts the resource manager migrator on the database `rmdb` with `icmrm` as the name of the resource manager web application.

To stop all applications selectively, enter the `etc/rc.cmrproc stop dbname rmapname application` command, where `dbname` is the database name on which these processes are running; `rmapname` is the name of the resource manager web application; and `application` is the resource manager stand-alone process that you want to start. For example, `/etc/rc.cmrproc stop rmdb icmrm RMMigrator` stops the resource manager migrator on the database `rmdb` with `icmrm` as the name of the resource manager web application.

Asynchronous Recovery utility overview

Content Manager includes an automatic scheduled process called the Asynchronous Recovery utility. 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 deletes of items that were 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 EIP toolkit and resource manager fails. The inconsistent state can be reconciled with the Asynchronous Transaction Reconciliation utility.

Another important result of running this utility is to cleanup 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 Version 8.1 or later 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. It should be understood that resource deletion is a multiple step process. On the 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. An EIP/CM 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 (icmrm`del`)
- The transaction reconciliation utility (icmrm`tx`)

Configuring the asynchronous recovery utility

The asynchronous recovery standalone utilities use the `icmpreenv.sh` (for AIX and Solaris) or the `icmpreenv.bat` (for Windows) for specifying the WebSphere directories when installing the resource manager. These files, found in the `%ICMR00T%/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 5. Using these files is a change from the Content Manager Version 8.1 Asynchronous Recovery utilities, where the `rmpath` and `DB2Instance` were 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 procedure on a Solaris Operating Environment system

To run the two Asynchronous Recovery utilities:

1. From a command prompt, enter: `cd /opt/IBMi/cm/bin.`
2. Enter `icmrmde1.sh` to run the deletion reconciliation utility.
3. Enter `icmrmtx.sh` to run the transaction reconciliation utility.

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(s) used by the resource manager through its defined device managers. Any of these components could fail and require a restoration via a backup that could be out of synch 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 RM/LS validation utility generates reports that describe discrepancies between the library server and the resource manager. The RM/V 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 tool or browser to view or manipulate the utility output files. Content Manager installs the XML DTD required by the validation utility output files.

Configuring the validation utilities

This section explains how to modify the two utility files with information specific to your Content Manager system. The shell scripts and batch files that invoke the validation utilities are located in the `bin` directory in the resource manager installation directory.

Modifying the scripts

The validation utilities are located in the `bin` directory in the resource manager installation directory. You type `icmrmlsval.sh` or `icmrmlsval.bat` to run the RM/LS validation utility. You type `icmrmvval.sh` or `icmrmvval.bat` to start the RM/V validation utility.

The validation utility creates and drops a temporary DB2 table. The environment script requires the resource 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`.

Logging: By default, the validation utilities log to a file named `icmrm.validator.log` file in the WebSphere logs directory. You can modify the level of information logged and the location of the output in the `icmrm_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 `icmrm_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/icmrm.ear  
| /icmrm.war/icmrm_validator_logging.xml
```

| On Solaris, the default path is:

```
| /opt/WebSphere/AppServer/installedApps/icmrm.ear  
| /icmrm.war/icmrm_validator_logging.xml
```

| On Windows, the default path is:

```
| x:\WebSphere\AppServer\installedApps\icmrm.ear  
| \icmrm.war\icmrm_validator_logging.xml
```

| Working with the resource manager/library server validation utility

| The RM/LS 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
| icmrm\sva1.sh or icmrm\sva1.bat.

| The utility requires input parameters that are described in Table 22. Both dashes (-)
| and forward slashes (/) are handled as the parameter separator. The parameter
| tags are supported in both lower and upper case.

| *Table 22. RM/LS 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 was 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 RM/LS 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 RM/LS reports

The base file names of the reports are "icmrmlsvalYYMMDDHHMMSS_" + *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 timestamp allows the administrator to run the utility multiple times without overwriting the output files. Examples of default names with the default report type are:

- icmrmlsval20020531123456_ORPHAN.xml
- icmrmlsval20020531123456_NOTINRM.xml
- icmrmlsval20020531123456_SIZEMISMATCH.xml
- icmrmlsval20020531123456_COLLECTIONMISMATCH.xml
- icmrmlsval20020531123456_DATEMISMATCH.xml

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.

The resource manager volume validation utility

The RM/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 a restore data on a volume after a volume crash. The utility will help you to verify that the data was restored correctly. The resource manager must be running when you use the utility. **Tip:** Use the 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 CM, the scanning for orphaned files could be extremely time consuming and might produce a large quantity of false positives.

The RM/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 RM/volume utility

The RM/volume validation utility is `icmrmvolval.sh` or `icmrmvolval.bat`. To start the utility, navigate to the `bin` directory in the resource manager home directory.

The RM/Volume program uses specific input parameters (see Table 23). Both dashes (-) and forward slashes (/) are handled as the parameter separator. The parameter tags are supported in both lower and upper case.

Table 23. RM/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.

Table 23. RM/Volume validation utility parameters (continued)

Parameter	Description
-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 timestamp allows the administrator to run the utility multiple times without overwriting the output files. Examples of default names with the default report type are:

- icmrmvolval20020531123456_FILENOTFOUND.xml
- icmrmvolval20020531123456_SIZEMISMATCH.xml

File not found

Entries are added to the FILENOTFOUND report if an object is in the resource manager database but it was not found on the volume recorded in the database. A file is considered "not found" if the volume's device manager either reported that the file did not exist, or reported that the file had 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.

Chapter 7. Managing user access

A user cannot access the Content Manager system without a user ID, password, or a privilege set. Before creating users and assigning them privileges, however, you must decide who will have access to the system and what their jobs require. You do not want users having the right to delete an object when they do not understand the ramifications of deleting that object. On the other hand, you do not want to prevent users from doing their jobs by not giving them the correct privileges. So, before assigning users privileges, you need to determine the types of tasks each job requires.

When users create an object in the Content Manager system, they must define the access that other users will have to that object. Users who create an object must define who can access the object and what operations can be done to the object. This definition is what is known to the Content Manager system as an access control list, or an ACL.

Creating user IDs and passwords

If you want a user ID that you define in the system administration client to also be used for DB2 authentication, then the user ID must follow the DB2 naming rules. The DB2 naming rules apply for user IDs that you want to use for either super administrators or connect user IDs. You cannot use the following words:

- USERS
- ADMINS
- GUESTS
- PUBLIC
- LOCAL
- Any SQL reserved word listed in the SQL Reference.

You cannot begin a user ID with the following characters:

- SQL
- SYS
- IBM

You can use the following characters:

- A through Z **Restriction:** some operating systems allow case-sensitive user IDs and passwords. Check your operating system documentation to see if it allows for case-sensitivity.
- 0 through 9
- #
- \$

Restriction: User IDs cannot exceed 30 characters.

Understanding DB2 administration authority

When logging on to the system administration client, you have two levels of authentication: one at the database level and another at the product level. Administrators have two classifications when you enable the administrative domains feature: super administrators and sub-administrators. In general, only super administrators have access to the system administration client.

Super administrators must have DB2 privileges: of db2admin, that is, full administrative privileges to DB2, are required. This user ID has to be defined in the operating system with the db2admin privilege. The password for this operating system ID is used to connect to DB2 and to log on to the library server. The password defined for the library server is not used. Content Manager privileges: This user ID is defined in the library server with full Content Manager administration privileges ("AllPrivs") to do all administration activities.

Sub-administrators do not require DB2 privileges. Sub-administrators manage only certain sections of the library server, therefore, sub-administrators log on to the system administration client one of two ways:

- If the user ID is an operating system user ID, then the password in the operating system is used to connect to DB2 and to log on to the library server.
- If the user ID is not an operating system user ID, then the user ID and password pair encrypted in the cmbfedenv.ini (for Enterprise Information Portal) or cmbicmenv.ini (for Content Manager) is used to connect to DB2 and the user ID and password provided in the Logon window is used to log on to the library server.

For more information about logging onto the library server, see the next section.

Sub-administrators also need the Content Manager privileges. They need the Domain Administrative privilege for all sub-domain administration activities.

Connecting to DB2 using the INI files

Each entry in the INI file contains the name of a library server and a pair of encrypted user ID and password for connecting to DB2. This encrypted user ID (known as connect user ID) and password are defined at the time you install the product. The connect user ID must be different than the system administrator's user ID. Enterprise Information Portal uses cmbfedenv.ini for connecting to DB2 and Content Manager uses cmbicmenv.ini. The default connect user ID is ICMCONCT. During installation, the passwords for the library server and the resource manager are contained in three places: The cmbicmenv.ini file contains the user ID and password to access the library server. The operating system defines access to the database where the library server and resource manager reside. The ICMRM.properties file contains the resource manager user ID and password

If the INI file is used, that is, the user ID is not an operating system user ID, then both the user ID and the connect user ID in the INI file must exist in the library server.

The connect user ID must be defined in the library server and operating system. It does require the UserDB2Connect privilege. To change the connect user ID and password in the INI file, select **Tools --> Change Database ID/password** from the administration client window.

Changing the library server and system administrator's password to the resource manager

If you need to change the password to the resource manager, then you need to change the password for the log on of the library server to the resource manager and the system administrator's password to the resource manager. **Important:** When changing the passwords for the log on of the library server and system administrator to the resource manager, complete the following steps in order:

1. Log on to the system administration client.
2. Expand the Resource Manager tree.
3. Click the resource manager that you want to modify and expand its tree.
4. Click Server Definitions and select Properties. The Server Panel window opens.
5. Change the password in the Password field.
6. Click OK.
7. Right-click the resource manager that you expanded (from step 2) and select Properties. The Resource Manager Properties window opens.
8. Change the password in the Password field and click OK.

Changing the database access passwords

If you need to change the database access passwords, you need to change the operating system password for the database connection and the ICMRM.properties file so that the resource manager can identify the new password.

To change the operating system password for the database connection, perform the following steps:

1. Depending on your operating system, navigate to the Users and Passwords utility.
2. Click ICMRM.
3. Select Set Password.
4. Enter the new password.

To change the ICMRM.properties file, complete the following steps:

1. Open the ICMRM.properties file. The default location is:
`X:\WebSphere\AppServer\installedApps\icrmr.ear\icrmr.war\WEB-INF\classes\com\ibm\mm\icrmr\ICMRM.properties` where *X* is the location of the drive in which you installed the Content Manager.
2. Change the DBPassword to match the operating system password.
3. Save the ICMRM.properties file.

After you change the database password, the database needs to either be restarted, or, you can let it issue two or three errors until it resets itself.

For detailed instructions about changing the passwords and other fields for a resource manager in the system administration client, see the system administration online help.

Importing users from LDAP

LDAP supports managing a user's ID and password at an enterprise level, rather than on a system-by-system basis. Content Manager makes use of three LDAP technologies: IBM Directory (known as IBM SecureWay Directory in previous versions) Windows 2000 Active Directory, and Lotus Domino Directory Notes

Address Book (NAB). The user password resides on the LDAP server. When a user logs on to Content Manager or , the user ID and password are authenticated and the user ID's specific privileges are checked by the user profile in the Content Manager database. LDAP might have been enabled during your Content Manager installation. If LDAP was not enabled during installation, you can activate it at any time.

To enable LDAP, select **Start → Programs → IBM Content Manager for Multiplatforms → LDAP User ID Import Scheduler** and then launch the system administration client. Bring up the LDAP Configuration window (Tools --> LDAP Configuration). Select the Enable LDAP user import and authentication check box and provide the LDAP server information on the Server page.

After you enable LDAP, you can import users by clicking the LDAP button in the New User window. This allows the users from the LDAP server to be selectively imported into Content Manager . Alternatively, you can import users in groups using the LDAP User ID Import Scheduler utility. During logon, the library server automatically connects to the LDAP server to authenticate the user. If the LDAP server is not able to verify the user's password for any reason, the authentication fails.

You can modify the LDAP server configuration by going to the main system administration client window and clicking **Tools -> LDAP Configuration**. You can also change your current LDAP server by going to the LDAP User Registry Import Utility from the Start --> Programs --> IBM Content Manager for Multiplatforms 8.2--> LDAP User ID Import Scheduler . For information about planning for LDAP, see *Planning and Installing Your Content Management System*. For information about how to configure LDAP server information in the system administration windows, see the system administration client online help.

For information about planning for LDAP, see *Planning and Installing Your Content Management System*. For information about how to implement LDAP, see the system administration client online help.

Introducing privileges

The administration client provides privilege groups, privilege sets, and individual privileges. If you administer a combined Content Manager/EIP system, the privileges are common to both parts of the client. The privileges that are built into the client can help you streamline the

Privilege group

A privilege group is a collection of user tasks for the purpose of helping administrators create new privilege sets or use roles in the Privilege set dialog.

Privilege sets

Privilege sets are a collection of user roles.

Privilege

A privilege represents a user action. For example,

Example 1 - privileges: You want to assign the privilege ClientScan and ClientImport to a group of users who typically use a client to only scan and import documents into Content Manager. If you had multiple users who typically performed that task, you would create one user ID (user1, for example). Then you would associate the privileges ClientScan and ClientImport with the user ID User1.

Then you would assign User1 to a Group named Group1. When any end user typed user1 to log in to their client and access Content Manager, that user would only be able to scan and import documents.

Example 2 - privilege groups: You have a group of experienced end users who require the privileges to access all the typical client tasks. You would create a user Id (for example, user2). Then you would assign user2 to a group (for example group2). Then you would associate the privilege group named ClientTaskAll to user1. When any end user typed user2 to log in to their client and access Content Manager, that user would be able to perform all the tasks contained in the privilege group named ClientTaskAll.

Example 3 - privilege sets: You have a group of users who require read-only access. You would create a user Id (for example, user3). Then you would assign user3 to a group (for example group3). Then you would associate the privilege set named ClientUserReadOnly to user3. When any end user typed user3 to log in to their client and access Content Manager, that user would be able to perform only the tasks contained in the privilege set named ClientUserReadOnly.

Creating privilege sets

When you plan your Content Manager system configuration, you must also decide who will have access to your system and how much access these users will have to the objects on your system. The Content Manager system defines access through privileges.

A privilege grants the right to access a specific object in a specific way. Privileges include rights such as creating, deleting, and selecting objects stored in a system. A group of privileges assigned to a user is a privilege set.

Your first task in managing access is to create privilege sets for users. A *privilege set* identifies the tasks or actions that a user can perform. Privilege sets combine privileges and are tailored for certain types of users. For example, you might want one set of administrators to manage your document routing server and another set of administrators to manage a domain. When an administrator logs on, Content Manager checks the administrator's privilege set.

The system administration client has a number of predefined privileges that you can group together into a privilege set. You then assign the privilege sets that you create to individual users. You cannot assign a privilege set to a user group.

Creating privilege groups

Privilege groups are like user groups for users. You create a privilege group to put similar privileges together to easily find the privileges you want to include in a privilege set. For example, if you have two privileges that you assign to almost every user in your system, instead of searching through the many privileges you have each time that you create a privilege set, you group these two basic privileges into a privilege group called BasicPrivs.

Assigning a privilege set to a user

The system administration client has a number of predefined privileges that you can group together into a privilege set. You then assign the privilege sets you create to individual users. You cannot assign a privilege set to a user group.

You can create privilege names, but you cannot create the privilege itself. You need to work with the system programmer to create any privileges that are not yet defined to the system administration client.

You can use the privilege sets that come with Content Manager, or you can create your own.

Assigning a user ID a grant privilege set

To prevent users from creating a user ID with more privileges than they have, Content Manager has implemented the use of a grant privilege set. When you assign a user ID with a grant privilege set, you give them authority to create user IDs within the limits of their granted privileges. For example, you can give a user ID a set of system administration privileges to manage a domain. You might, however, want to ensure that the user ID does not have the privilege to create users. So, when you create this user ID, in the grant privilege set field, you would select "Noprivs". In effect, the user ID can manage the domain but cannot create users for that domain.

Assigning users to resource managers

To allow users to access a specific resource manager, you assign a resource manager to a domain that users have access to. For more information about assigning resource managers to domains, see "Assigning a resource manager to a domain" on page 99.

Assigning users to collections

To allow users to access to collections, you assign a collection on a resource manager to a domain that users have access to. For more information about assigning collections to domains, see "Assigning a collection to a domain" on page 99.

Creating user groups

Often, users with the same job description have the same or similar tasks, and therefore, the same access to objects on your system. You can group users with common access needs together in a user group. You cannot nest user groups.

A user group is solely a convenience grouping of individual users with similar tasks. You do not assign a user group a privilege set. Each user in a user group has his or her own privilege set. A user group makes it easier to create access control lists for objects in your system.

If you have domains enabled, before you assign a user ID to a group, check to see if that user group is in a specific domain or the PUBLIC domain (see "Administering domains" on page 98 for more information about domains). Make sure that the user group is in the domain that you want your user ID to be in. If you want to create a user ID specifically for a domain, you can click **New User** within the User Group window. You can then add the user that you create to the user group, and ensure that the user is in the same domain.

Creating access control lists

You provide users with the privileges that they need to accomplish their tasks. Objects, on an individual basis, have certain access control issues.

An access control list (ACL) is a list consisting of one or more individual user IDs or user groups and their associated privileges. You use ACLs to control user access to objects in the Content Manager system. The objects that can be associated with access control lists are: the data objects stored by users, item types and item type subsets, worklists, and processes.

Privilege sets define the individual user's maximum ability to use the system, an ACL restricts an individual user's access to an object. An ACL that has a privilege not defined by a user's privilege set does not grant the user with that privilege. Only users that have that privilege can use that privilege on an object. An ACL limits user access, it does not grant more access. Access control lists provide another level of security when managing a system.

Assigning a privilege set to an access control list

Each user ID that you add to an access control list (ACL) needs a privilege set associated with it. The user ID and privilege set define which users have access to an object and what kind of access they have to that object.

Users cannot access any object unless they are on the ACL. To add a user or user group to an ACL, you need to select a user ID and a privilege set for the ACL and click **Add**. For each defined ACL, you will find the user IDs and groups listed in the Access Control List window. You can modify this table by adding and removing user IDs and groups. For more information about creating or modifying an ACL, see the system administration client online help.

Creating domains

A domain is a section of a library server that one or more administrators manage. Domains consist of user IDs, user groups, privilege sets, access control lists, resource managers, and SMS collections. Domains are not visible to users, so what you name your domains will only have meaning to you and the system administrators who manage them. Users do not know that you have limited them to a part of the library server, meaning that they only know about items within that domain.

Domains limit administrative and user access to a subsection of the library server. An administrator with full privileges to the library server can delegate limited administrative privileges to another administrator. The administrator with full privileges, a super administrator, has access to all sections of a library server while an administrator with limited privileges, a subadministrator, has access to only a section of the library server.

Domains restrict the access a subadministrator has to access control lists (ACLs). Only super administrators can create ACLs that subadministrators can use to either add or delete user IDs and user groups. Subadministrators cannot create, update, or delete ACLs.

A subadministrator might share different combinations of the super administrator responsibilities but only for their domain. By creating domains and assigning administrators to manage those domains, the super administrators can delegate subtasks while concentrating on the overall system and managing it efficiently as the subadministrators manage users and tasks specific to their domain.

Before you enable domains, consider the following conditions:

- You cannot disable domains

- Resource managers, collections, user IDs, and user groups can exist in only one domain at a time
- Privilege sets and access control lists can exist in more than one domain at a time
- Except for the PUBLIC (shared) domain, domains do not overlap
- Any object created in the super administrative domain cannot be moved, whether if it is system generated or user created.

To enable domains, go to the file menu, select **Tools** → **Administrative Domains** and then select **Enable Administrative Domains**. You need to restart the system administration client for the domains to take effect. For specific instructions about how to configure your library server for domains, see the system administration client online help.

Administering domains

Depending on your privilege set, you administer either the whole library server or a specific domain. An administrator who has full access to the library server is a super administrator. A subadministrator has full access to the objects in a specific domain.

Each type of administrator has the ability to create, retrieve, update, and delete the objects in their domains, including users and collections. Subadministrators can see and retrieve objects only in their domain and list or retrieve in the PUBLIC, or shared, domain.

Accessing domains

Subadministrators cannot change the domain of an object. They can, however, access the contents of their own domain and list or retrieve any object in the PUBLIC, or shared, domain.

Super administrators have access to all domains on the library server. They can create an object and assign it to a domain. Some objects, such as privilege sets and ACLs, only they can create for subadministrators to use.

Subadministrators can only do create, retrieve, update, and delete (CRUD) for any objects in their domain.

Assigning a user to a domain

When you create a user ID, you have the choice to assign it to a domain, or leave it in the default domain. You can change the domain of the user ID at a later time through user properties.

A user ID can have access to only one domain at a time. You cannot add a user to the PUBLIC, or shared, domain.

Only super administrators have the authority to create domains and assign users to those domains. A domain can have more than one subadministrator, but only the super administrator can define who those administrators are by giving them system administration privileges within a privilege set. The **Grant privilege set** field in the New User or User Properties window will indicate which administrative privileges a subadministrator has within a domain.

Assigning a user group to a domain

Assigning a user group to a domain changes the domain designated for each user ID in that user group. A user ID can have access to only one domain at a time. So, any user ID included in a group that you assign is also moved to the new domain.

A user group name cannot be in only one domain at a time. You can assign the user group into the PUBLIC, or shared, domain.

Assigning a privilege set to a domain

Any user ID that you add to a domain must also have an associated privilege set. If you do not include the associated privilege sets, then the users cannot perform their tasks. The best place to store privilege sets to make them available to any user is the PUBLIC, or shared, domain.

Assigning a resource manager to a domain

You can restrict user access to certain resource managers by assigning them to a specific domain. When you define a new resource manager for a library server to access, you have the option to select a domain.

The default for all resource managers is PUBLIC. If you do not want everyone to have access to the resource manager, you need to assign it to a domain. If you do not see a domain that you can assign the resource manager to, you can still define the resource manager and then create the domain you need. After you have the appropriate domain defined, open the resource manager properties and select the domain.

Assigning a collection to a domain

You can restrict user access to a certain collection on a resource manager by assigning it to a specific domain. If the resource manager is in the PUBLIC domain, you can assign a collection to any other defined domain. If the resource manager, however, is defined to a specific domain already, then you cannot assign the collection to another domain, even if you want to assign the collection to the PUBLIC domain.

A user needs access to the resource manager to access the collections on it, so you cannot restrict access to the resource manager without imposing the same restrictions to the collections on it.

Moving a user from one domain to another

You might find reason to remove certain users from one domain and add them to another. Consider using the **Description** field in the User definition window as a way to remember which user groups a user is grouped in. It might make this task a little easier.

Important: This task is very time consuming and can result in problems with accessing the system if you do not do it right. You must be a super administrator to change the domain of a user.

Follow these steps carefully:

1. Find all of the groups that the user belongs to.
2. For all the groups the user belongs to, either move these groups to the PUBLIC domain, or remove the user from all the groups.

3. Move any resource manager associated with this user to the PUBLIC domain, followed by all of the collections for each resource manager that you move to the target domain.
4. Create, *do not move*, all of the privilege sets associated with the user in the target domain, if they are not already in the target domain.
5. Create, *do not move*, all of the access control lists associated with this user, if they are not in the target domain.
6. Move the user to the target domain by opening the user's Properties and changing the user's domain.
7. **Optional:** You can move the groups and resource manager that you moved in steps 1 on page 99, 2, and 3 from the PUBLIC domain to the target domain, but you can only do so if there are no more users remaining in the source domain who are associated with the groups and resource managers that you move. Otherwise, the groups and resource managers need to stay in the PUBLIC domain to allow sharing for users in different domains.

Reminder: At no time can a user be in the PUBLIC domain. Users cannot be shared.

Moving a user group from one domain to another

Important: This task can result in problems with accessing the system if you do not do it right. You must be a super administrator to change the domain of a user group.

Follow these steps to move a user group to a different domain:

- If the user group is empty, delete the group from its current domain then recreate the group and assign it to the target domain.
- If the user group is not empty, follow these steps:
 1. Find all of the users that belong to this group.
 2. Delete the group from its current domain, which will delete all of the users.
 3. Recreate the group and assign it to the target domain.
 4. Add all of the users to this newly created group.

Moving a resource manager from one domain to another

You must be a super administrator to change the domain of a resource manager.

To move a resource manager to another domain, follow these steps:

- If the resource manager contains no collections, move the resource manager to the target domain by opening its properties and changing the domain to the target domain.
- If the resource manager contains collections, follow these steps:
 1. Move the resource manager to the PUBLIC domain.
 2. Move the collections to the target domain by opening Properties and selecting the target domain.
 3. Move the resource manager to the target domain by opening Properties and selecting the target domain.

Moving a collection from one domain to another

You must be a super administrator to change the domain of a collection.

Follow these steps to move a collection from one domain to another:

1. Find out the resource manager the collection belongs to.
2. Move the associated resource manager to the PUBLIC domain.
3. Move the collection to the target domain by opening Properties and selecting the target domain.
4. Move the resource manager to the target domain by opening Properties and selecting the target domain.

Moving a privilege set from one domain to another

Because privilege sets can reside in multiple domains, you can add them to the target domain without moving them.

Moving an access control list from one domain to another

Because access control lists can reside in multiple domains, you can add them to the target domain without moving them.

Chapter 8. Managing databases

The information related to the objects stored in the resource manager is maintained both in the library server and the resource manager. It is conceivable that data related to objects stored in the resource manager and library server can become unsynchronized. It is crucial to keep the data synchronized between the resource manager and library server. The resource manager provides utilities to help you synchronize the data.

You also need to manage the objects that are stored in the database. The resource manager schedules when objects need to migrate and replicate. You can do schedule migration and replication of objects when you configure resource managers for your system.

Optimizing server databases

A table can become fragmented after many updates, causing performance to deteriorate. Queries take longer because index entries in the library server and resource manager are no longer synchronized with the actual data in the database tables.

You can synchronize the data in the index with the database tables by running the `reorgchk` command in DB2. The `reorgchk` command gathers and compares both the index and the table statistics and recommends tables to reorganize. Most of the time, performance improves simply by running `reorgchk`, but if not, then you must reorganize the database tables.

When you reorganize tables, you remove empty spaces and arrange table data efficiently. Reorganizing tables take a lot more time than running `reorgchk`. Do not reorganize tables when you expect a lot of server activity because performance will be slow. DB2 locks any data in a table that it is currently being reorganized.

Consider the following factors to determine when to reorganize your table:

- The volume of insert, update, and delete activity.
- Running `reorgchk` does not improve the performance of queries.

Though not advisable, you can reorganize a table at any time. If you update tables often, then you want to reorganize periodically, for example, once a month. If you do not manage the DB2 database tables, you need to work with the DB2 administrator for access or to coordinate when to run `reorgchk` and reorganize tables.

Optimizing a DB2 database

If you manage the DB2 database, then you need to run periodic table updates using `reorgchk`. You can find instructions about how to update database tables in the *DB2 Command Reference* (click **Start** → **Programs** → **IBM DB2** → **Information** → **DB2 Information** and type `reorgchk` in the search field). Use the *DB2 Command Reference* and the following instructions to check and update database tables:

1. Open a DB2 Command Window by clicking **Start → Programs → IBM DB2 → Command Window**. If you are not already connected to the database, connect to the database by entering `db2 connect to icmnlbdb` where `icmnlbdb` is the name of the database.
2. When you run `reorgchk`, you need to store the results in a file. This file, also known as a log file, contains the statistics you need to use to determine whether to reorganize a table. For example, if you want to update all tables, you need to enter:


```
db2 reorgchk update statistics on table all > out.txt
```

where `out.txt` is the name of the log file.
3. Check the `Reorg` column in your log file. DB2 displays 1 to 3 asterisks (***) in the `Reorg` column when it detects a table to reorganize. The asterisks determine the urgency of reorganizing a table.
4. Note the schema name and table name (the first two columns). You use these two names to reorganize tables. For example, a schema name could be `icmadmin` or `sysibm` and a table name could be `icmstnlkeywords` or `sysindexes`.
5. Use the *DB2 Command Reference* to see how to reorganize tables. For example, you might type `db2 reorg Table sysibm.sysindexes` to reorganize the `sysindex` table.
6. Do another `reorgchk` to see if you have any more tables to reorganize. Complete the previous steps to reorganize any other tables you want.
7. When you finish reorganizing database tables, you need to rebind all packages using the `db2rbind` command. You do not need to be connected to the database for this step. Enter:


```
db2rbind icmnlbdb /l report.txt
```

in the DB2 Command Window, where `icmnlbdb` is the name of the database and `report.txt` is the name of the log file that contains the results. **Important:** You need a user ID and password if you plan to update schema that do not belong to you. Also, the user ID and password must have DB2 administrative authority to complete this task.

8. Check your log file to see the results. Another way that you can check the success of a rebind is by using the DB2 Control Center:
 - a. Open the Control Center by clicking **Start -> Programs -> IBM DB2 -> Control Center**.
 - b. Go to the database against which you ran `db2rbind`.
 - c. In the database, go to **Application objects -> Packages**.
 - d. Check the columns, Last bind date and Last bind time. The date and time indicate when you last had DB2 rebind all the packages.

For more information about `reorgchk` and other DB2 commands, see the *DB2 Command Reference*. For a more detailed understanding of reorganizing and rebinding DB2 database tables, see the *DB2 System Administration Guide*.

Removing entries from the events table

When you use the Content Management system administration client, the library server records item and document routing related functions in the events table, `ICMSTSYSADMEVENTS` or `ICMSTITEMEVENTS`.

The events table grows with each logged event. To reduce the size of the events table, you can remove the expired and unused events from the table. The EventCode column in the events table indicates the classification of events as the following values:

Value Definition

1–200 System administration function event codes

200–900

Item, document routing, and resource management function event codes

1000+ Application event codes

You can delete events from the events table by performing either of these following tasks:

- To delete an event for a system administration function from a library server, connect to your database and use the following SQL command:

```
delete from ICMSTSYSADMEVENTS where eventcode <=200 and Created <
2002-01-01-12.00.00.000000
```

- To delete an event for an item function from a library server, connect to your database and use the following SQL command:

```
delete from ICMSTITEMEVENTS where eventcode <=600 and Created <
2002-05-01-12.00.00.000000
```

Consult your DB2 administrator for help with connecting to the appropriate database.

To reclaim the file system space after you delete the events, run the database reorganization utility on the library server database and then stop the database instance.

Migrating objects

Managing your object storage is essential to keeping your Content Manager system efficient. It saves time and money because it moves the less frequently used objects to a slower device, allowing the faster, more expensive devices, to handle objects in high demand.

After you decide where you want to place objects, you need to decide if the objects need to stay in that location, or if they need to move to another location. This path is known as a migration policy.

Each migration policy belongs to a collection of objects, called collections. When you create your migration policy, you decide how long to store a collection in a storage system. You use a migrator schedule to check the migration policy for collections for which time has expired. When the migrator schedule begins, and the time for the collection in its current storage class has elapsed, then the migration policy moves the collection to the next storage class.

Creating a migration policy

To create a migration policy in the system administration client, you need to already have the storage classes defined. See the system administration client online help about how to create storage classes. Creating a migration policy and defining the migrator schedule automates the migration of objects so you do not have to manually monitor migration.

To create a migration policy, expand the section for the resource manager that you want to manage the collection migration. You need to right-click **Migration Policies** and select **New**. Any field marked with an asterisk (*) is required for you to complete. It is possible to create a migration policy with just a name, but you cannot use it until you add the storage classes and their retention periods.

If you decide to migrate the collection to a remote storage system, you need to select **Move to remote storage class**. Each storage class is associated with one or more storage systems. The remote storage class needs to be already created. The storage class you designate as the remote storage class identifies a resource manager and collection with which it belongs.

After you create your migration policy, you need to assign it to a collection. If you do not assign it to a collection, then it does not get used, even when you only have one collection defined in your resource manager.

The system administration client online help gives you more detail about how to create a migration policy.

Setting up remote migration

If you plan to migrate a collection to another resource manager, you need to create a storage class in the current resource manager that directs the collection to the remote resource manager. Create a storage class for each remote resource manager that you plan to use for remote migration. See the system administration client online help about how to create storage classes.

When you create a remote storage class, you need to select a **Resource manager** and a **Collection**. These two pieces of information can immediately supply the location of where objects need to go. For more about collections, see “Collections” on page 67.

When you decide to include a storage class indicating that a collection needs to move to a remote location, it needs to be the last step in the migration policy. To specify the remote location, you need to select **Move to remote storage class** and pick a remote storage class that you created on the current resource manager.

Changing the date of migration

When you migrate objects, you need to tell Content Manager how long you want to retain a collection, and when to check for migrating the collections.

The first task is to decide how long you want to retain a collection. You designate the retention period when you create a migration policy. You have two choices, either you keep the collection in a storage system for a certain number of days, or, you keep the collection in a storage system forever. You can change the amount of time by viewing the properties of a migration policy, selecting the storage class that you want to change, and clicking **Edit**. In the window that opens, you can change the amount of time of the storage class to its new time. For more details about changing the date of migration for the migration policy, see the system administration client online help.

The second task is to configure the migrator schedule for your resource manager. You can find the migrator schedule by completing the following steps:

1. Expand **Resource Managers**.

2. Expand the resource manager that contains the migration policy in which you want to schedule.
3. Right click **Configurations**.
4. Click the Migrator Schedule tab.

You must decide when you want the migration of objects to occur. You have two choices in the panel: **Every day** or **Specific day**. The time you select launches the migrator schedule to check if the retention period of the collection in a migration policy has expired. If the time has expired, then the resource manager moves the collection to the next storage class listed in the migration policy.

Migrating and purging the VideoCharger Server media objects at regular intervals

To configure how often to migrate and purge media objects to the Multimedia Archive at regular intervals, complete the following steps:

1. Expand **Resource Managers** in the system administration client main window.
2. Expand the resource manager that manages the VideoCharger server that contains the schedule for migrating and purging.
3. Right-click **Configuration**. The Resource Manager Configuration window opens.
4. Click the **Cycles** tab.
5. Under Cycles, select **Enabled** for **Purger** and **Migrator**.
6. Set how often to purge and migrate by typing the corresponding **Hours** and **Minutes**.
7. Under Batches (files), set how many files you want to migrate simultaneously by typing a number for **Stager** and **Migrator**. The default is 50 files.
8. Click **OK** to save your changes and close the window.

Content Manager then automatically starts, enables, and stops both Stager and Migrator during the intervals you specified.

Chapter 9. Managing document routing

Your task is to create and manage the document routing processes defined in your company's business plan. In earlier Content Manager, document routing was known as workflow. An instance of document routing is called a process.

Document routing processes can contain work baskets and collection points. Processes determine the flow of work to complete, so when you create the work baskets and collection points of each process, focus on the tasks that users need to accomplish. Privilege sets and access control lists determine who does the task.

Document routing moves documents or folders from one work node to another. A work node, which is a general term for work baskets and collection points, is a step within a process at which items wait for actions to be taken by end users or applications, or items move ahead automatically.

Each work node belongs to one or more worklists. A worklist contains a list of work packages based on priority or state (such as suspend or notify). A work package contains the information that a user needs to complete a task. The user is unaware of a work package because the user works on the item it references, not on the work package itself. A work package contains a set of information such as priority, state, resume time, process, and ItemID being routed. Content Manager supports a complex process, allowing you to create processes that determine what route a work package takes based on the actions or non-actions of end users or applications.

You need to create and manage processes. As part of creating processes, you define work baskets, collection points, and worklists. You must change processes to reflect changes to your business. You might have to force work through to the next step in a process, terminate a process, or suspend a process.

You can set conditions to do these tasks automatically, but sometimes you must update these conditions. For example, instead of suspending a document for 10 days, you want to suspend it for seven days. To update this task, you must call an API to suspend a process and pass in the suspend time as an input parameter.

Defining a process

A *process* is a series of steps through which an item is routed. A process contains at least one start node, one action, and one end node. (You can use these one-step processes to create ad hoc processes.) Processes can have as many steps as you want.

To define a new process, you must have:

- A name for your process
- A predefined ACL
- Predefined work nodes and collection points

You can create a variety of processes. You can create serial processes that take work from start to finish without any deviations. You can also create dynamic processes that allow you to direct work through different routes depending on the actions that you specify.

Content Manager provides two ways to create a process: **Continue** and **Escalate**. These two choices do not have any meaning other than providing you with the opportunity to create a process that branches. For example, if you want an insurance claim to go from one node to another, you can select **Continue** as the path it takes. Then, you create a point where the action of the user dictates where the work package goes next. If the insurance claim is approved, it will continue on the **Continue** path. If it is rejected, you can create a path that branches off the **Continue** path by using **Escalate**. If you do not like either one of these labels, you can create your own label by typing it in the field provided. Your label will now appear as one of the choices in the drop down menu.

Defining work baskets

Each step in a process corresponds to a real-world task, like verifying a record or rejecting an insurance application. Work baskets contain work packages. A work package contains the location of a document or folder in a database and its priority. A work basket does not perform any actions on the content, rather, it is an indicator of where a work package is in a process. When you assign an ACL to a work basket, you give access to users who can perform actions on the work packages contained in that work basket.

A work basket is more than just a virtual basket that has a pile of work stacked in it. You decide what functions a work basket requires to get a work package to where it needs to go. You need to specify, through dynamic link libraries (DLLs), what tasks work packages complete upon entering and leaving a work basket. You can also specify what a work package must do when a work basket cannot contain it by using a DLL on the condition that the work basket is overloaded. Your DLLs must contain the host name of the machine where a DLL resides.

To define a work basket, you need:

- A name for your work basket
- A predefined ACL
- The full location, host name and directory of any DLLs that you plan to use

You need to define a DLL and function name to apply when a package enters, exits, and when a work basket becomes too full. These actions define how a work basket gets a work package where it needs to go. Specifying an overload DLL and function name allows you to direct a work package to the appropriate handler so that the work package does not get deleted.

Defining collection points

A *collection point* is a special work node that waits for external documents to be collected in a folder, but does not correspond to a business task. It merely collects required documents and send them to another work node when it either completes a folder or the time allotted to wait for the documents has expired.

To define a collection point, you need:

- A name for your collection point
- A predefined ACL
- The full location, host name and directory of any DLLs that you plan to use
- A list of required item types to complete a folder

A collection point is strictly used in document routing processes. It has nothing to do with resource manager collections.

Adding a work basket or collection point to a process

You can add a work basket or collection point to a process at any time. You might update a process because the way an enterprise performed a process at one time has changed, or no longer exists. If you want to change the location of a work basket or collection point, you must delete it from the process and then add it to the new location.

After you define a work basket or collection point, open a new or existing process, and click **Add**. From the pull-down menu, select your work basket or collection point. If what you want to include is not there, then return to the main window and click **Refresh**.

For more information about how to add, delete, and update a work node, consult the online help.

Branching in a process

You can create a process that allows users to direct work packages to a work node or collection point depending on the decisions they make. For example, you might want an insurance claim to go into one work basket if the claimant's last name begins with A through M and to another work basket if a claimant's name begins with N through Z. You could create your own actions by adding `A_Z` and `N_Z` to the **Selection** field in the New Process window. Then, when the user entering the claimant's name sends it off to the next work basket, the work package follows the path depending on the claimant's last name.

Restriction: You must work with your programmer to create a client application that supports the recognition of different actions. Certain actions might not be supported by the client that your end users use.

Branching is just like creating any process in the system administration client, however, you must define more than one action in the **Selection** field. The action that you select defines a link from one work node to another. A branching occurs when one process has two or more actions in the **Selection** field. The action indicates where a work package is going.

You can create your own actions that the system administration client keeps in the drop down menu. You can use these in several processes or create actions unique to each process.

Ad hoc routing processes

Ad hoc routing processes allow you to remove a document or folder from one process and put it in another.

Your programmer can help you to create ad hoc routing processes. Ad hoc routing processes consist of a series of one-step processes. You can define these processes, but only the programmer can use the appropriate APIs to direct work from one process to another.

For an ad hoc routing process, you need at least two lines in the New Process window. **Start** and **End** are virtual work baskets. They only indicate a work process has started or ended. If you try to save a new process with only these two labels, then you get an error.

Defining worklists

A worklist is one or more work nodes from which a user obtains a list of work or the next work package. A worklist spans all work nodes and collection points, regardless of the process.

You need to assign work nodes and collection points to a worklist and give the worklist an access control list (ACL). The ACL filters out the users that can access the work nodes and collection points. The ACL of the work nodes and collection points further restricts access to the work packages in them. For example, an insurance underwriter and an underwriter assistant can have access to the same worklist, but, based on their privileges and the ACL of the work nodes and collection points, the underwriter will have a different list of work packages than the underwriter assistant.

For more information about creating worklists, see the system administration client online help.

Defining work packages

A work package is a set of information such as priority, state, resume time, and Item ID being routed. It is used to relate an item to a work node. You do not create work packages. Work packages are created by the system with information from the user who starts a process. The user logs on to Content Manager and proceeds to start a process. Content Manager prompts the user to specify the process, the item ID that will use this process, and the item priority. Content Manager takes this information and creates a work package that proceeds through the process.

For more information on starting a process, see the eClient information.

Creating folders for a process

You can use an item type to create a folder by using the folder semantic type. You can create a folder called Customer that contains a customer's insurance policy, police reports, and any claim that customer has filed.

For more information about creating item types, see "Item types" on page 15, or view the online help.

Updating a process

You can update a process at any time, even when a process is in use. Any changes you make immediately affect the process. For example, if you create a work basket that a work package has not reached yet, then, when the work package arrives at the new work basket, it uses the work basket as if it was always there. If you add a collection point in a place where the work package has already passed, the work package will continue on its route as if the work basket or collection point had been there all along. The work package is not affected by any changes in nodes it has already passed through.

Deleting a process

If you want to delete a process, you must wait until all work packages on the process are complete. You cannot delete a process when it is in use nor can you prevent anyone from starting a process that you want to delete. You cannot determine when a process is in use because you cannot view who is using the process. You can attempt to delete the process until the system allows you to delete it.

To delete a process, select the process name from the main system administration window, right-click it and click **Delete**.

ICM library server event table log

Table 24 explains the information you see for event codes 1 - 208. These event codes are the system administration and logon event codes. You can disable the logging of events 1 - 85 and 500 - 522 by setting the SysAdminEventFlag value in ICMSTSYSCONTROL table to 0. To enable the logging, set the value to 1.

Table 24. System administration and logon event codes

Column Name	Data Type	Attribute
Event Code	Integer	NOT null
Created	Timestamp	NOT null
User ID	Char(32)	NOT null
EventData1	Varchar(254)	nullable
EventData2	Varchar(254)	nullable
EventData3	Varchar(254)	nullable
EventData4	Varchar(254)	nullable
EventData5	Varchar(254)	nullable

Table 25 explains the data provided for item events. You can set the type of logging for this table by opening the ICMSTITEMTYPEDEFS table and defining the ItemEventFlag value. The following value definitions perform the corresponding logging functions:

- 0 False, CRUD disabled (default)
- 1 Create, enabled
- 2 Read, enabled
- 3 Create & Read, enabled
- 4 Update, enabled
- 7 Create, Read & Update, enabled
- 8 Delete, enabled
- 15 Create, Read, Update & Delete, enabled

Table 25. Item events table

Column Name	Data Type	Attribute
Event Code	Integer	NOT null
Created	Timestamp	NOT null
Item ID	Char(26)	NOT null
User ID	Char(32)	NOT null
EventData1	Varchar(254)	nullable
EventData2	Varchar(254)	nullable
EventData3	Varchar(254)	nullable
EventData4	Varchar(254)	nullable
EventData5	Varchar(254)	nullable

Table 26 describes the data you might see in the event log. Event codes 1 - 608 are reserved for the library server functions. Event code 1000 and above are for the user-defined functions. Currently, the library server has five different event code classifications:

- System administration functions: 1 - 85
- Logon functions: 201 - 208
- Item functions: 301 - 404
- Workflow system administration functions: 500 - 522
- Resource objects and document routing events: 530 - 608

Table 26. Library server event logging table

EventCode	EventData1	EventData2	EventData3	EventData4	EventData5
1 ADD USER	User ID	User Name	User Privilege Set	Grant Privilege Set	Default Item ACL
2 UPDATE USER	User ID	User Name	User Privilege Set	Grant Privilege Set	Default Item ACL
3 DELETE USER	User ID	N/A	N/A	N/A	N/A
4 ADD USER GROUP	Group User ID	Group Name	N/A	N/A	N/A
5 UPDATE USER GROUP	Group User ID	Group Name	N/A	N/A	N/A
6 DELETE USER GROUP	Group User ID	N/A	N/A	N/A	N/A
7 ADD ACL	ACL Code	ACL Name	Language Code	N/A	N/A
8 UPDATE ACL	ACL Code	ACL Name	Language Code	N/A	N/A
9 DELETE ACL	ACL Code	Language Code	N/A	N/A	N/A
11 INCREMENTAL UPDATE ACL	SP Name	Action	Privilege Set Code	Privilege Definition Code	N/A
12 ADD LANGUAGE	Language Code	Language Name	N/A	N/A	N/A
13 UPDATE LANGUAGE	Language Code	Language Name	N/A	N/A	N/A
14 DELETE LANGUAGE	Language Code	N/A	N/A	N/A	N/A
15 ADD PRIVILEGE	SP Name	Action	Privilege Definition Code	Privilege Definition Name	Privilege Description
16 UPDATE PRIVILEGE	SP Name	Action	Privilege Definition Code	Privilege Definition Name	Privilege Description
17 DELETE PRIVILEGE	SP Name	Action	Privilege Definition Code	N/A	N/A
19 UPDATE SYS CONTROL PARM	ACL Binding Level	Library ACL Code	Public Access Enable	Default ACL Choice	SMS Choice
21 ADD ATTRIBUTE	Language Code	Attribute ID	Attribute Name	Attribute SQL Type	Attribute Length
22 UPDATE ATTRIBUTE	Language Code	Attribute ID	Attribute Name	Attribute SQL Type	Attribute Length
23 DELETE ATTRIBUTE	Language Code	Attribute ID	N/A	N/A	N/A
24 ADD ATTRIBUTE GROUP	Language Code	Attribute Group	Attribute Group Name	N/A	N/A

Table 26. Library server event logging table (continued)

EventCode	EventData1	EventData2	EventData3	EventData4	EventData5
25 UPDATE ATTRIBUTE GROUP	Language Code	Attribute Group	Attribute Group Name	N/A	N/A
26 DELETE ATTRIBUTE GROUP	Language Code	Attribute Group	N/A	N/A	N/A
27 ADD COLLECTION NAME	RM Code	SMS Collection Code	User ID	Prefetch Indicator	SMS Collection Name
29 DELETE COLLECTION NAME	RM Code	SMS Collection Code	N/A	N/A	N/A
33 ADD COMPONENT	Component Type ID	Component Type Name	Component Type Description	Item Type ID	Parent Component Type ID
34 UPDATE COMPONENT	Component Type ID	Component Type Name	Component Type Description	User ID	N/A
35 DELETE COMPONENT	Component Type ID	Component Type Name	Component Type Description	N/A	N/A
36 BUILD COMPONENT TYPE	Schema Name	Component Type Name	Table Name	Item Type Name	Parent Component Type Name
37 ADD ITEM TYPE	Item Type ID	Item Type Name	Item Type Description	N/A	N/A
38 UPDATE ITEM TYPE	Item Type ID	Item Type Name	Item Type Description	N/A	N/A
39 DELETE ITEM TYPE	Item Type ID	Item Type Name	Item Type Description	N/A	N/A
40 GET ITEM TYPE	Number of Item Type ID	Detail	Number of Privilege Code	N/A	N/A
41 ADD KEYWORD CLASS	Keyword Class	Keyword Code	Language Code	Keyword Name	Keyword Description
42 ADD KEYWORD CODE	Keyword Class	Keyword Code	Language Code	Keyword Name	Keyword Description
43 UPDATE KEYWORD CODE	Keyword Class	Keyword Code	Language Code	Keyword Name	Keyword Description
44 DELETE KEYWORD CODE	Keyword Class	Keyword Code	N/A	N/A	N/A
45 ADD LINK TYPE	Keyword Class	Keyword Code	Language Code	Keyword Name	Keyword Description
46 UPDATE LINK TYPE	Keyword Class	Keyword Code	Language Code	Keyword Name	Keyword Description
47 DELETE LINK TYPE	Keyword Class	Keyword Code	N/A	N/A	N/A
48 ADD PRIVILEGE SET	SP Name	Action	Privilege Set Code	Privilege Definition Code	N/A
49 UPDATE PRIVILEGE SET	SP Name	Action	Privilege Set Code	Privilege Set Name	Privilege Set Description
50 DELETE PRIVILEGE SET	SP Name	Action	Privilege Set Code	N/A	N/A

Table 26. Library server event logging table (continued)

EventCode	EventData1	EventData2	EventData3	EventData4	EventData5
51 ADD COMPONENT VIEW	Component View ID	Component Type ID	Item Type ID	View Display Name	User ID
52 UPDATE COMPONENT VIEW	Component View ID	Component View Name	User ID	N/A	N/A
53 DELETE COMPONENT VIEW	Component View ID	Component View Name	Language Code	N/A	N/A
54 ADD ITEMTYPE VIEW	Item View ID	Item Type ID	ACL Code	Language Code	User ID
55 UPDATE ITEMTYPE VIEW	Item View ID	Item Type View Name	Language Code	N/A	N/A
56 DELETE ITEMTYPE VIEW	Item View ID	Language Code	N/A	N/A	N/A
57 ADD EVENT TYPE	Keyword Class	Keyword Code	Language Code	Keyword Name	Keyword Description
58 UPDATE EVENT TYPE	Keyword Class	Keyword Code	Language Code	Keyword Name	Keyword Description
59 DELETE EVENT TYPE	Keyword Class	Keyword Code	N/A	N/A	N/A
60 ADD SEMANTIC TYPE	Keyword Class	Keyword Code	Language Code	Keyword Name	Keyword Description
61 UPDATE SEMANTIC TYPE	Keyword Class	Keyword Code	Language Code	Keyword Name	Keyword Description
62 DELETE SEMANTIC TYPE	Keyword Class	Keyword Code	N/A	N/A	N/A
63 ADD XDO TYPE	XDO Class ID	Attribute Group ID	XDO Class Name	N/A	N/A
64 UPDATE XDO TYPE	XDO Class ID	Attribute Group ID	XDO Class Name	N/A	N/A
65 DELETE XDO TYPE	XDO Class ID	N/A	N/A	N/A	N/A
66 ADD PRIVILEGE GROUP	Language Code	Privilege Group Code	Privilege Group Name	Privilege Group Description	Number of Privileges
67 UPDATE PRIVILEGE GROUP	Language Code	Privilege Group Code	Privilege Group Name	Privilege Group Description	N/A
68 DELETE PRIVILEGE GROUP	Language Code	Privilege Group Code	N/A	N/A	N/A
69 ADD SET ACL	ACL Code	User ID	User Kind	Privilege Set Code	N/A
70 UPDATE SET ACL	ACL Code	User ID	User Kind	Privilege Set Code	N/A
71 DELETE SET ACL	ACL Code	User ID	N/A	N/A	N/A
72 ADD COMPONENT ATTR	SP Name	Language Code	Component Type ID	Number of Attributes	N/A
73 ADD INDEX ON COMPONENT	SP Name	Action	Index Name	Component Type ID	Number of Attributes
74 DELETE INDEX ON COMPONENT	SP Name	Action	Index Name	N/A	N/A

Table 26. Library server event logging table (continued)

EventCode	EventData1	EventData2	EventData3	EventData4	EventData5
75 ADD ITEM RELATION	Source Item Type ID	Target Item Type ID	N/A	N/A	N/A
76 UPDATE ITEM RELATION	Source Item Type ID	Target Item Type ID	N/A	N/A	N/A
77 DELETE ITEM RELATION	Source Item Type ID	Target Item Type ID	N/A	N/A	N/A
78 ADD ADMIN DOMAIN	Domain ID	Domain Name	Language Code	N/A	N/A
79 UPDATE ADMIN DOMAIN	Domain ID	Domain Name	Language Code	N/A	N/A
80 DELETE ADMIN DOMAIN	Domain ID	Language Code	N/A	N/A	N/A
81 ADD DOMAIN ACL	Domain ID	Number of ACL	N/A	N/A	N/A
82 DELETE DOMAIN ACL	Domain ID	Number of ACL	N/A	N/A	N/A
83 ADD DOMAIN PRIVILEGE SET	Domain ID	Number of Privilege Set	N/A	N/A	N/A
84 DELETE DOMAIN PRIVILEGE SET	Domain ID	Number of Privilege Set	N/A	N/A	N/A
85 CHANGE USER PASSWORD	User ID	Expiration Date	User Name	N/A	N/A
201 LOGON	User ID	Event Time ddhmmssmsms	Application	Password Flag	N/A
202 LOGOFF	User ID	N/A	N/A	N/A	N/A
203 LOGON INVALID USERID	User ID	Event Time	Application	N/A	N/A
204 LOGON INVALID PASSWORD	User ID	Event Time	Application	N/A	N/A
205 LOGON MAX USERS REACHED	User ID	Event Time	Application	N/A	N/A
206 LOG MAX USER ERROR REACHED	User ID	Event Time	Application	N/A	N/A
207 LOGON PASSWORD CHANGED	User ID	Event Time	Application	N/A	N/A
208 LOGON USER EXIT ERROR	User ID	Event Time	Application	N/A	N/A
301 CREATE ITEM	Item Type Name	N/A	N/A	N/A	N/A
302 UPDATE ITEM	Old Version ID	New Version ID	Item Type Name	N/A	N/A
303 DELETE ITEM	Version ID	N/A	N/A	N/A	N/A
305 UPDATE OBJECT DATA	Version ID	Ext Object Name	Resource Length	N/A	N/A
306 REINDEX ITEM	Item Type Name	N/A	N/A	N/A	N/A

Table 26. Library server event logging table (continued)

EventCode	EventData1	EventData2	EventData3	EventData4	EventData5
401 GET ITEM	Component ID	Component View Name	Item Type View Name	N/A	N/A
402 ADD AUTO LINK	Target Item Type Name	Source Item Type Name	N/A	N/A	N/A
403 UPDATE AUTO LINK	Target Item Type Name	Source Item Type Name	N/A	N/A	N/A
404 DELETE AUTO LINK	Target Item Type Name	Source Item Type Name	N/A	N/A	N/A
500 ADD WORKFLOW ACTION	Action Code	Action Name	Language Code	Predefine Action	N/A
501 UPDATE WORKFLOW ACTION	Action Code	Action Name	Language Code	Predefine Action	N/A
502 DELETE WORKFLOW ACTION	Action Code	Action Name	Language Code	N/A	N/A
503 ADD WORKFLOW ACTIONLIST	SP Name	Action	Action List	N/A	N/A
504 UPDATE WORKFLOW ACTIONLIST	SP Name	Action	Action List	Action List Name	Action List Description
505 DELETE WORKFLOW ACTIONLIST	SP Name	Action	Action List	N/A	N/A
506 ADD WORKFLOW DIAGRAM	SP Name	Action	Diagram ID	Diagram Name	Diagram Description
507 UPDATE WORKFLOW DIAGRAM	SP Name	Action	Diagram ID	Diagram Name	Diagram Description
508 DELETE WORKFLOW DIAGRAM	SP Name	Action	Diagram ID	N/A	N/A
509 CHECKIN DIAGRAM	SP Name	Action	Diagram ID	Diagram Name	Diagram Description
510 CHECKOUT DIAGRAM	SP Name	Action	Diagram ID	Diagram Name	Diagram Description
511 ADD WORKLIST	Work List Code	ACL Code	Language Code	Work List Name	Work List Description
512 UPDATE WORKLIST	Work List Code	ACL Code	Language Code	Work List Name	Work List Description
513 DELETE WORKLIST	Work List Code	N/A	N/A	N/A	N/A
514 ADD COLLECTION POINT	SP Name	Action	Process ID	Collection Activity ID	WF Starter ID

Table 26. Library server event logging table (continued)

EventCode	EventData1	EventData2	EventData3	EventData4	EventData5
515 UPDATE COLLECTION POINT	SP Name	Action	Process ID	Collection Activity ID	WF Starter ID
516 DELETE COLLECTION POINT	SP Name	Action	Process ID	N/A	N/A
517 ADD WORKFLOW EVENT	Activity ID	Process ID	WF Starter ID	N/A	N/A
518 UPDATE WORKFLOW EVENT	Activity ID	Process ID	WF Starter	N/A	N/A
519 DELETE WORKFLOW EVENT	Activity ID	N/A	N/A	N/A	N/A
520 ADD DIAGRAPMPROMPT	SP Name	Action	Diagram ID	Number of Prompts	N/A
521 UPDATE DIAGRAPMPROMPT	SP Name	Action	Diagram ID	Number of Prompts	N/A
522 DELETE DIAGRAPMPROMPT	SP Name	Action	Diagram ID	Number of Prompts	N/A
539 SETUP RM FLAG	RM Name	N/A	N/A	N/A	N/A
540 ADD WORKFLOW ACTIONLIST CODE	SP Name	Action	Action List	Action List Name	Action List Description
600 DR START PROCESS	Process Name	Work Node Name	N/A	N/A	N/A
601 DR ROUTE ITEM	Process Name	Work Node Name	Next Work Node Name	N/A	N/A
602 DR END PROCESS	Process Name	Work Node Name	N/A	N/A	N/A
605 DR OVERLOAD	Process Name	Work Node Name	Number of Work Packages Currently in Work Node	N/A	N/A
606 DR WORKNODE PASSTHROUGH	Process Name	Work Node Name	N/A	N/A	N/A
607 ADD REPLICA RULES	Source RM Name	Target SMS Collection Code	Number of Replica Rules	N/A	N/A
608 DELETE REPLICA RuLES	Source RM Name	Target SMS Collection Code	Number of Replica Rules	N/A	N/A
609 UPDATE REPLICA RULES	Source RM Name	Target SMS Collection Code	Number of Replica Rules	N/A	N/A

Accessibility features

This product includes a number of features that make it more accessible for people with disabilities. These features include:

- The ability to operate all features using the keyboard instead of the mouse.
- Support for enhanced display properties
- Compatibility with assistive technologies
- Compatibility with operating system accessibility features
- Accessible documentation formats

Keyboard input and navigation

The following features are available for keyboard input and navigation:

Keyboard input

You can use the keyboard instead of a mouse to operate the product.

Menu items and controls provide access keys that allow you to activate a control or select a menu item directly from the keyboard. These keys are self-documenting; the access keys are underlined on the control or menu where they appear.

Keyboard focus

In Windows-based systems, the position of the keyboard focus is highlighted, indicating which area of the window is active and where your keystrokes will have an effect.

Response time adjustments

In Windows-based systems, you can adjust response times through your control panel.

Features for accessible display

The clients have a number of features that enhance the user interface and improve accessibility for users with low vision. These enhancements include support for high-contrast settings and customizable font properties.

High-contrast mode

The clients support the high-contrast mode option that is provided by the operating system. This feature supports a higher contrast between background and foreground colors.

Font settings

In Windows-based systems, you can specify display settings that determine the color, size, and font for the text in menus and dialog windows. The client allows you to select the font for the document list.

Non-dependence on color

You do not need to distinguish between colors in order to use any function of this product.

Compatibility with assistive technologies

The clients are compatible with screen reader applications such as Narrator and Via Voice. The clients have properties required for these accessibility applications to make onscreen information available to visually impaired users.

Accessible documentation

Documentation for this product is available in PDF format. You can convert the PDF files to HTML or text using free tools available from Adobe at access.adobe.com. This allows users to view documentation according to the display preferences set in their browsers. It also allows the use of screen readers and other assistive technologies.

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.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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

IBM	DisplayWrite	PowerPC
400	e-business	PTX
Advanced Peer-to-Peer Networking	HotMedia	QBIC
AIX	Hummingbird	RS/6000
AIXwindows	ImagePlus	SecureWay
APPN	IMS	SP
AS/400	Micro Channel	VideoCharger
C Set ++	MQSeries	Visual Warehouse
CICS	MVS/ESA	VisualAge
DATABASE 2	NetView	VisualInfo
DataJoiner	OS/2	WebSphere
DB2	OS/390	
DB2 Universal Database	PAL	

Approach, Domino, Lotus, Lotus 1-2-3, Lotus Notes and SmartSuite are trademarks or registered trademarks of the Lotus Development Corporation in the United States, other countries, or both.

Intel and Pentium are trademarks or registered trademarks of Intel Corporation in the United States, other countries, or both.

Microsoft, Windows, and Windows NT are registered 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.

Glossary

This glossary defines terms and abbreviations specific to this system. Terms shown in *italics* are defined elsewhere in this glossary.

A

abstract class. An object-oriented programming *class* that represents a concept; classes derived from it represent implementations of the concept. You cannot construct an object of an abstract class; that is, it cannot be instantiated.

access control. The process of ensuring that certain functions and stored *objects* can be accessed only by authorized users in authorized ways.

access control list. A list consisting of one or more user IDs or user groups and their associated *privileges*. You use access control lists to control user access to *items* and *objects* in the Content Manager system.

accessory script. A *CGI script* that processes SEARCH, POST, PUT, or DELETE requests. The accessory scripts process requests that are not explicitly mapped to a CGI script named on an EXEC directive.

action list. An approved list of the actions, defined by a system administrator or some other *workflow coordinator*, that a user can perform in a *workflow* or document routing process.

address. The unique code assigned to each device or workstation connected to a network. See also *IP address*.

admission control. The process used by the server to ensure that its bandwidth needs are not compromised by new asset requests.

ADSM. See *Tivoli Storage Manager*.

aggregate bandwidth. Total throughput, in megabits per second, that moves through a server or server subsystem.

alias. In the *Internet*, a name assigned to a server that makes the server independent of the name of its host machine. The alias must be defined in the *domain name server*.

American National Standard Code for Information Interchange (ASCII). The standard code, using a coded character set consisting of 7-bit coded characters (8 bits including parity check), that is used for information interchange among data processing systems, data communication systems, and associated equipment. The ASCII set consists of control characters and graphic characters.

analog video. Video in which the information that represents images is in a continuous-scale electrical signal for amplitude and time.

API. See *application programming interface*.

application programming interface (API). A software interface that enables applications to communicate with each other. An API is the set of programming language constructs or statements that can be coded in an application program to obtain the specific functions and services provided by the underlying licensed program.

application server. Software that handles communication with the client requesting an asset and queries of the Content Manager.

archive. Persistent storage used for long-term information retention, typically very inexpensive for each stored unit and slow to access, and often in a different geographic location to protect against equipment failures and natural disasters.

ASCII. See *American National Standard Code for Information Interchange*.

asset. A digital multimedia resource that is stored for later retrieval as requested by an application. An example of such a resource is a digitized video or audio file. An asset is stored as a file in a multimedia file system supported by the *data pump*.

asset group. An organizational grouping within the multimedia file system with similar characteristics. You can use an asset group to allocate resources of a *data pump*. For example, you could establish two asset groups representing distinct departments whose assets should be kept separate for security or billing purposes.

asymmetric video compression. In multimedia applications, the use of a powerful computer to compress a video so that a less powerful system can decompress it.

asynchronous transfer mode (ATM). A transfer mode in which the information is organized into cells; it is asynchronous in the sense that the recurrence of cells containing information from an individual user is not necessarily periodic. ATM is specified in international standards such as ATM Forum UNI 3.1.

attribute. A unit of data that describes a certain characteristic or property (for example, name, address, age, and so forth) of an item, and which can be used to locate that item. An attribute has a type, which indicates the range of information stored by that attribute, and a value, which is within that range. For

example, information about a file in a multimedia file system, such as title, running time, or encoding type (MPEG1, H.263, and so forth).

attribute group. Convenience grouping of one or more *attributes*. For example, Address might include the attributes Street, City, State, and Zip.

audio. The sound portion of a video signal.

Audio/Video Interleaved (AVI). A RIFF (*Resource Interchange File Format*) file specification that permits audio and video data to be interleaved in a file. The separate tracks can be accessed in alternate chunks for playback or recording while maintaining sequential access on the file device.

Audio-Video Subsystem (AVS). File format for files that can contain video and audio data, video-only data, audio-only data, or image data (a single still image). The Audio-Video Subsystem format is supported by the ActionMedia II MMPM/2 Media Control interface.

AVI. See *Audio/Video Interleaved*.

AVS. See *Audio-Video Subsystem*.

B

background. The conditions under which low priority, non-interactive programs are run.

bandwidth. (1) The difference, expressed in *Hertz*, between the highest and the lowest frequencies of a range of frequencies. (2) In *asynchronous transfer mode* (ATM), the capacity of a virtual channel, expressed in terms of peak cell rate (PCR), sustainable cell rate (SCR), and maximum burst size (MBS). (3) A measure of the capacity of a communication transport medium (such as a TV cable) to convey data.

base attributes. A set of indexes that is assigned to each *object*. All Content Manager objects have base *attributes*.

baseband. A frequency band that uses the complete bandwidth of a transmission.

batch. (1) An accumulation of data to be processed. (2) A group of records or data processing jobs brought together for processing or transmission.

binary large object (BLOB). A sequence of bytes with a size ranging from 0 bytes to 2 gigabytes. This string does not have an associated code page and character set. Image, audio, and video objects are stored in BLOBs.

bitmap. (1) A representation of an image by an array of bits. (2) A pix map with a depth of one bit plane.

BLOB. See *binary large object*.

block. A string of data elements recorded or transmitted as a unit. The elements can be characters, words, or physical records. Disk device drivers currently use a block size of 32 KB or 256 KB to write to the disk.

broadband. A frequency band divisible into several narrower bands so that different kinds of transmissions (such as voice, video, and data) can occur at the same time. See *baseband*.

bus. A facility for transferring data between several devices located between two end points, only one device being able to transmit at a given moment.

C

cache. A special-purpose buffer, smaller and faster than main storage, used to hold a copy of data that can be accessed frequently. Use of a cache reduces access time, but might increase memory requirements. See also *resource manager cache* and *LAN cache*.

caching proxy server. A proxy server that can store the documents it retrieves from other servers in a local *cache*. The catching proxy server can then respond to subsequent requests for these documents without retrieving them from other servers, a process that can improve response time.

cardinality. The number of rows in a database table.

category. See *item type*.

CGI. See *Common Gateway Interface*.

CGI script. A computer program that runs on a Web server and uses the *Common Gateway Interface (CGI)* to perform tasks that are not usually done by a Web server (for example, database access and form processing). A CGI script is a CGI program that is written in a scripting language such as Perl.

child component. Optional second or lower level of a hierarchical *item type*. Each child component is directly associated with the level above it.

CIF. See *common interchange file*.

CIU. See *common interchange unit*.

class. In object-oriented design or programming, a model or template that can be instantiated to create objects with a common definition and therefore, common properties, operations, and behavior. An object is an instance of a class.

client. A computer system or process that requests a service of another computer system or process that is typically referred to as a server. Multiple clients can share access to a common server.

client application. An application written with the Content Manager APIs to customize a user interface.

Client Application for Windows. A complete object management system provided with Content Manager and written with Content Manager APIs. It supports document and folder creation, storage, and presentation, processing, and access control. You can customize it with user exit routines and partially invoke it with APIs.

client/server. In communications, the model of interaction in distributed data processing in which a program at one site sends a request to a program at another site and awaits a response. The requesting program is called a client; the answering program is called a server.

codec. A processor that can code analog audio or video information in digital form for transmission, and decode digital data back to analog form.

collection. A group of objects with a similar set of management rules.

Common Gateway Interface (CGI). A standard for the exchange of information between a Web server and programs that are external to it. The external programs can be written in any programming language that is supported by the operating system on which the Web server is running. See *CGI script*.

common interchange file (CIF). A file that contains one ImagePlus Interchange Architecture (IPIA) data stream.

common interchange unit (CIU). The independent unit of transfer for a common interchange file (CIF). It is the part of the CIF that identifies the relationship to the receiving database. A CIF can contain multiple CIUs.

component. Generic term for a *root component* or a *child component*.

compressed audio. A method of digitally encoding and decoding several seconds of voice quality audio per single videodisc frame. This increases the storage capability to several hours of audio per videodisc. Sometimes referred to as still frame audio or sound over still.

compressed video. A video resulting from the process of digitally encoding and decoding a video image or segment using a variety of computer techniques to reduce the amount of data required to represent the content accurately.

compression. The process of eliminating gaps, empty fields, redundancies, and unnecessary data to shorten the length of records or blocks.

connection manager. A Content Manager component that helps maintain connections to the library server, rather than starting a new connection for each query. The connection manager has an application programming interface.

container. An element of the user interface that holds objects. In the *folder manager*, an *object* that can contain other folders or documents.

content class. See *MIME type*.

controller. The functional component responsible for resource management (load balancing and admission control). The controller communicates with one or more *data pumps* to initiate and terminate connections to clients.

cursor. A named control structure used by an application program to point to a specific row within some ordered set of rows. The cursor is used to retrieve rows from the set.

D

data format. See *MIME type*.

data pump. The combination of the disks that hold the data and the networking hardware and software required to deliver assets to clients.

data rate. The rate at which data is transmitted or received from a device. Interactive applications tend to require a high data rate, while batch applications can usually tolerate lower data rates.

datastore. Generic term for a place (such as a database system, file, or directory) where data is stored.

data striping. Storage process in which information is split into blocks (a fixed amount of data) and the blocks are written to (or read from) a series of disks in parallel.

data transfer rate. The average number of bits, characters, or blocks per unit time passing between corresponding equipment in a data transmission system.

Notes:

1. The rate is expressed in bits, characters, or blocks per second, minute, or hour.
2. Corresponding equipment should be indicated; for example, modems, intermediate equipment, or source and sink.

DCA. See *document content architecture*.

DCE. See *Distributed Computing Environment*.

decode. To convert data by reversing the effect of some previous encoding.

decompression. Process of restoring compressed data to its original state, so that it can be used again.

destager. A function of the Content Manager *resource manager* that moves objects from the *staging area* to the first step in the object's *migration policy*.

device driver. Software used to manage a specific device. Other software uses the device driver as the interface to the device for reading, writing, and control functions.

device manager. In a Content Manager system, the interface between the *resource manager* and one or more physical devices.

digital. Pertaining to data in the form of digits.

digital audio. Audio tones represented by machine-readable binary numbers rather than by analog recording techniques.

digital video. Video in which the information (usually including audio) is encoded as a sequence of binary digits. The information is usually compressed. It can be stored and transported just as any other digital information. Viewing digital video involves decompressing the video data, converting it to an analog form, displaying the video on a monitor, and playing the sound through an amplifier and speakers.

digitize. To convert analog video and audio signals into digital format.

digitized image. An image derived from a scanning device or a digitizing card with a camera.

Distributed Computing Environment (DCE). The Open Software Foundation (OSF) specification (or a product derived from this specification) that assists in networking. DCE provides such functions as authentication, directory service (DS), and remote procedure call (RPC).

document. An *item* that can be stored, retrieved, and exchanged among Content Manager systems and users as a separate unit. An item with the document *semantic type* is expected to contain information that forms a document, but does not necessarily imply that it is an implementation of the Content Manager document model.

An item created from a document classified item type (a specific implementation of the Content Manager document model), must contain document parts. You can use document classified item types to create items with either the document or folder semantic type.

Document parts can include varied types of content, including for example, text, images, and spreadsheets.

document content architecture (DCA). An architecture that guarantees information integrity for a document being interchanged in an office system network. DCA

provides the rule for specifying form and meaning of a document. It defines revisable form text (changeable) and final form text (unchangeable).

document root directory. The primary directory where a Web server stores accessible documents. When the server receives requests that do not point to a specific directory, it tries to serve the request from this directory.

document routing process. In Content Manager a sequence of *work steps*, and the rules governing those steps, through which a *document* or *folder* travels while it is being processed.

document type definition (DTD). The rules that specify the structure for a particular class of XML documents. The DTD defines the structure with elements, attributes, and notations, and it establishes constraints for how each element, attribute, and notation can be used within the particular class of documents. A DTD is analogous to a database schema in that the DTD completely describes the structure for a particular markup language.

domain. That part of a computer network in which the data processing resources are under common control.

domain name. In the *Internet* suite of *protocols*, a name of a host system. A domain name consists of a sequence of subnames separated by a delimiter character.

domain name server. In the *Internet* suite of *protocols*, a server that responds to queries from clients for name-to-address and address-to-name mappings as well as for other information.

dotted decimal notation. The syntactical representation of an IP address. The 4 bytes of the address are written as four decimal numbers separated by periods (dots), for example, 9.37.83.123.

DTD. See *document type definition*.

E

element. An *object* that the *list manager* allocates for an application.

encode. To convert data by using a code in such a manner that reconversion to the original form is possible.

Ethernet. A 10-Mbps baseband local area network that allows multiple stations to access the transmission medium at will without prior coordination, avoids contention by using carrier sense and deference, and resolves contention by using collision detection and transmission.

Extensible Markup Language (XML). A standard metalanguage for defining markup languages that was derived from, and is a subset of, SGML. XML omits the more complex and less-used parts of SGML and makes it much easier to write applications to handle document types, author and manage structured information, and transmit and share structured information across diverse computing systems. The use of XML does not require the robust applications and processing that is necessary for SGML. XML is being developed under the auspices of the World Wide Web Consortium (W3C).

External Data Representation (XDR). A standard, developed by Sun Microsystems, Incorporated, for representing data in machine-independent format.

F

F-Coupler (frequency coupler). A physical device that merges broadband analog signals with digital data on an IBM Cabling System using shielded twisted-pair wiring. The IBM F-Coupler separates analog signals and sends them from the IBM Cabling System to the workstation. The F-Coupler allows the IBM Cabling System to accommodate simultaneous analog video with data traffic on a token-ring network.

FDDI. See *Fiber Distributed Data Interface*.

feature. The visual content information that is stored in the image search server. Also, the visual traits that image search applications use to determine matches. The four QBIC[®] features are average color, histogram color, positional color, and texture.

Fiber Distributed Data Interface. An American National Standards Institute (ANSI) standard for a 100-Mbps LAN using optical fiber cables.

file name extension. An addition to a file name that identifies the file type (for example, text file or program file).

file system. In AIX, the method of partitioning a hard drive for storage. See also *multimedia file system*.

file system manager. The component that manages the multimedia file system.

File Transfer Protocol (FTP). In the *Internet* suite of *protocols*, an application layer protocol that uses *Transmission Control Protocol (TCP)* and Telnet services to transfer bulk-data files between machines or hosts.

firewall. (1) In communication, a functional unit that protects and controls the connection of one network to other networks. The firewall (a) prevents unwanted or unauthorized communication traffic from entering the protected network and (b) allows only selected

communication traffic to leave the protected network. (2) In equipment, a partition used to control the spread of fire.

folder. An *item* of any *item type*, regardless of classification, with the folder *semantic type*. Any item with the folder semantic type contains specific folder functionality that is provided by Content Manager, in addition to all non-resource item capabilities and any additional functionality available from an item type classification, such as *document* or resource item. Folders can contain any number of items of any type, including documents and subfolders. A folder is indexed by *attributes*.

folder manager. The Content Manager model for managing data as online documents and folders. You can use the folder manager APIs as the primary interface between your applications and the Content Manager content servers.

fps. Frames per second. The number of frames displayed per second.

fragment. The smallest unit of file system disk space allocation. A fragment can be 512, 1024, 2048, or 4096 bytes in size. The fragment size is defined when a file system is created.

frequency coupler. See *F-coupler*.

FTP. See *File Transfer Protocol*.

full-motion video. Video reproduction at 30 frames per second (*fps*) for NTSC signals or 25 *fps* for PAL signals.

G

gateway. A functional unit that interconnects two computer networks with different network architectures. A gateway connects networks or systems of different architectures. A bridge interconnects networks or systems with the same or similar architectures.

GB. See *gigabyte*.

gigabyte (GB). (1) For processor storage, real and virtual storage, and channel volume, 2^{30} , or 1 073 741 824 bytes. (2) For disk storage capacity and communications volume, 1 000 000 000 bytes.

H

handle. A character string that represents an object, and is used to retrieve the object.

Hertz (Hz). A unit of frequency equal to one cycle per second. In the United States, line frequency is 60 Hz or a change in voltage polarity 120 times per second; in

Europe, line frequency is 50 Hz or a change in voltage polarity 100 times per second.

history log. A file that keeps a record of activities for a *workflow*.

home page. The initial Web page that is returned by a Web site when you enter the address for the Web site in a Web browser. For example, if a user specifies the address for the IBM Web site, which is <http://www.ibm.com>, the Web page that is returned is the IBM home page. Essentially, the home page is the entry point for accessing the contents of the Web site.

host. A computer, connected to a network, which provides an access point to that network. A host can be a client, a server, or a client and a server simultaneously.

host name. In the *Internet* suite of *protocols*, the name given to a computer. Sometimes, host name refers to the fully qualified domain name; other times, it is used to mean the most specific subname of a fully qualified domain name. For example, if mycomputer.city.company.com is the fully qualified domain name, either of the following might be considered the host name:

- mycomputer.city.company.com
- mycomputer

HTML. See *Hypertext Markup Language*.

HTTP (Hypertext Transfer Protocol). In the *Internet* suite of *protocols*, the protocol that is used to transfer and display hypertext documents

HTTPd. See *HTTP daemon*.

HTTP daemon. A multithreaded Web server that receives incoming *Hypertext Transfer Protocol (HTTP)* requests.

HTTP method. An action used by the *Hypertext Transfer Protocol (HTTP)*. HTTP methods include GET, POST, and PUT.

Hypertext Markup Language (HTML). A markup language that conforms to the SGML standard and was designed primarily to support the online display of textual and graphical information that includes hypertext links.

Hz. See *Hertz*.

I

I frame (information frame). In video compression a frame that has been compressed independently of any other frames. Also referred to as a reference frame, intra frame, or still frame.

Image Object Content Architecture (IOCA). A collection of constructs used to interchange and present images.

index. To add or edit the attribute values that identify a specific *item* or *object* so that it can be retrieved later.

index class. See *item type*.

index class subset. In earlier Content Manager, a view of an *index class* that an application uses to store, retrieve, and display folders and objects.

index class view. In earlier Content Manager, the term used in the APIs for *index class subset*.

inline. In Content Manager, an object that is online and in a drive, but has no active *mounts*. Contrast with *mounted*.

i-node. In the AIX operating system, the internal structure that describes the individual files in the operating system; there is one i-node for each file. An i-node contains the node, type, owner, and location of a file. A table of i-nodes is stored near the beginning of a *file system*.

interactive video. Combining video and computer technology so the user's actions determine the sequence and direction the application takes.

interchange. The capability to import or export an image with its index from one Content Manager ImagePlus for OS/390 system to another ImagePlus system using a *common interchange file* or *common interchange unit*.

Internet. The worldwide collection of interconnected networks that use the *Internet* suite of *protocols* and permit public access.

Internet Protocol (IP). In the *Internet* suite of *protocols*, a connectionless protocol that routes data through a network or interconnected networks and acts as an intermediary between the higher protocol layers and the physical network.

intranet. A private network that integrates *Internet* standards and applications (such as Web browsers) with an organization's existing computer networking infrastructure.

IOCA. See *Image Object Content Architecture*.

IP. See *Internet Protocol*.

IP address. The unique 32-bit address that specifies the actual location of each device or workstation on the *Internet*. The address field contains two parts: the first part is the network address; the second part is the host number. For example, 9.67.97.103 is an IP address.

IP multicast. Transmission of an *Internet Protocol (IP)* datagram to a set of systems that form a single multicast group. See *multicast*.

ISO-9660. Format used for files on CD-ROM. Used with DOS.

isochronous. A communications capability that delivers a signal at a specified, bounded rate, which is desirable for continuous data such as voice and full-motion video.

item. In Content Manager, generic term for an instance of an *item type*. For example, an item might be a *folder, document, video, or image*.

item type. A template for defining and later locating like *items*, consisting of a *root component*, zero or more *child components*, and a classification.

item type classification. A categorization within an *item type* that further identifies the *items* of that item type. All items of the same item type have the same item type classification.

Content Manager supplies the following item type classifications: *folder, document, object, video, image, and text*; users can also define their own item type classifications.

J

JavaBeans™. A platform-independent, software component technology for building reusable Java components called “beans.” After they are built, these beans can be made available for use by other software engineers or can be used in Java applications. Using JavaBeans, software engineers can manipulate and assemble beans in a graphical drag-and-drop development environment.

Joint Photographic Experts Group (JPEG). (1) A group that worked to establish the standard for the compression of digitized continuous-tone images. (2) The standard for still pictures developed by this group.

JPEG. See *Joint Photographic Experts Group*.

K

Kb. See *Kilobit*.

KB. See *Kilobyte*.

Kbps. *Kilobits per second*.

key field. See *attribute*.

kilobit (Kb). (1) For processor storage, real and virtual storage, and channel volume, 210 or 1024 bits. (2) For disk storage capacity and communications volume, 1000 bits.

kilobyte (KB). (1) For processor storage, real and virtual storage, and channel volume, 210 or 1024 bytes. (2) For disk storage capacity and communications volume, 1000 bytes.

L

LAN. See *local area network*.

LAN cache. An area of temporary storage on a local *resource manager* that contains a copy of objects stored on a remote resource manager.

latency. The time interval between the instant at which an instruction control unit initiates a call for data and the instant at which the actual transfer of the data starts.

LBR. See *low bit rate*.

library client. The component of a Content Manager system that provides a low-level programming interface for the library system. The library client includes APIs that are part of the software developer’s kit.

library object. See *item*.

library server. The component of a Content Manager system that stores, manages, and handles queries on *items*.

link. A directional relationship between two *items*: the source and the target. You can use a set of links to model one-to-many associations. Contrast with *reference*.

local area network (LAN). A network in which a set of devices are connected to one another for communication and that can be connected to a larger network.

low bit rate (LBR). A generic term for an interleaved H.263/G.723 stream. Low bit rate streams range from 6.4 Kbps up to 384 Kbps.

M

machine-generated data structure (MGDS). (1) An IBM structured data format protocol for passing character data among the various Content Manager ImagePlus for OS/390 programs. (2) Data extracted from an image and put into general data stream (GDS) format.

management class. The term used in the APIs for *migration policy*.

Management Information Base (MIB). A collection of objects that can be accessed by means of a network management *protocol*.

maximum transmission unit (MTU). In *LANs*, the largest possible unit of data that can be sent on a given

physical medium in a single frame. For example, the MTU for *Ethernet* is 1500 bytes.

Mb. See *megabit*.

MB. See *megabyte*.

Mbps. *Megabits per second*.

MCA. See *Micro Channel architecture*.

media archiver. A physical device that is used for storing audio and video stream data. The VideoCharger is a type of media archiver.

media server. An AIX-based component of the Content Manager system that is used for storing and accessing video files.

megabit (Mb). (1) For processor storage, real and virtual storage, and channel volume, 220 or 1 048 576 bits. (2) For disk storage capacity and communications volume, 1 000 000 bits.

megabyte (MB). (1) For processor storage, real and virtual storage, and channel volume, 220 or 1 048 576 bytes. (2) For disk storage capacity and communications volume, 1 000 000 bytes.

method. In Java design or programming, the software that implements the behavior specified by an operation. Synonymous with member function in C++.

MGDS. See *machine-generated data structure*.

MIB. See *Management Information Base*.

MIB variable. A managed object that is defined in the *Management Information Base (MIB)*. The managed object is defined by a textual name and a corresponding object identifier, a syntax, an access mode, a status, and a description of the semantics of the managed object. The MIB Variable contains pertinent management information that is accessible as defined by the access mode.

Micro Channel Architecture (MCA). The rules that define how subsystems and adapters use the Micro Channel *bus* in a computer. The architecture defines the services that each subsystem can or must provide.

MIDI. See *Musical Instrument Digital Interface*.

migration. (1) The process of moving data and source from one computer system to another computer system without converting the data, such as when moving to a new operating environment. (2) Installation of a new version or release of a program to replace an earlier version or release.

migration policy. A user-defined schedule for moving *objects* from one *storage class* to the next. It describes the retention and class transition characteristics for a group of objects in a storage hierarchy.

migrator. A function of the *resource manager* that checks *migration policies* and moves objects to the next *storage class* when they are scheduled to move.

MIME type. An Internet standard for identifying the type of object being transferred across the Internet. MIME types include several variants of audio, image, and video. Each object has a MIME type.

Mixed Object Document Content Architecture (MO:DCA). An IBM architecture developed to allow the interchange of object data among applications within the interchange environment and among environments.

Mixed Object Document Content Architecture–Presentation (MO:DCA–P). A subset architecture of MO:DCA that is used as an envelope to contain documents that are sent to the Content Manager ImagePlus for OS/390 workstation for displaying or printing.

M-JPEG. See *Motion JPEG*.

MO:DCA. *Mixed Object Document Content Architecture*

MO:DCA–P. *Mixed Object Document Content Architecture—Presentation*

Motion JPEG (M-JPEG) . Used for animation.

mount. To place a data medium in a position to operate.

mounted. In Content Manager, an object that is online and in a drive, with active *mounts*. Contrast with *inline*.

Moving Pictures Expert Group (MPEG). (1) A group that is working to establish a standard for compressing and storing motion video and animation in digital form. (2) The standard under development by this group.

MPEG. See *Moving Pictures Expert Group*.

MTU. See *maximum transmission unit*.

multicast. Transmission of the same data to a selected group of destinations.

multimedia. Combining different media elements (text, graphics, audio, still image, video, animation) for display and control from a computer.

multimedia file system. A *file system* that is optimized for the storage and delivery of video and audio.

Multipurpose Internet Mail Extensions (MIME) . See *MIME type*.

Musical Instrument Digital Interface (MIDI). A *protocol* that allows a synthesizer to send signals to

another synthesizer or to a computer, or a computer to a musical instrument, or a computer to another computer.

N

name server. See *domain name server*.

National Television Standard Committee (NTSC). (1) A committee that sets the standard for color television broadcasting and video in the United States (currently in use also in Japan). (2) The standard set by the NTSC committee.

network table file. A text file that contains the system-specific configuration information for each node in a Content Manager system. Each node in the system must have a network table file that identifies the node and lists the nodes that it needs to connect to.

The name of a network table is FRNOLINT.TBL.

NTSC. See *National Television Standard Committee*.

O

object. Any digital content that a user can store, retrieve and manipulate as a single unit, for example, *JPEG* images, *MP3* audio, *AVI* video, and a text block from a book.

Object Linking and Embedding (OLE). A Microsoft specification for both linking and embedding applications so that they can be activated from within other applications.

object server. See *resource manager*.

object server cache. See *resource manager cache*.

OLE. See *Object Linking and Embedding*.

overlay. A collection of predefined data such as lines, shading, text, boxes, or logos, that can be merged with variable data on a page during printing.

P

package. A collection of related *classes* and interfaces that provides access protection and namespace management.

page pool. The area in the shared memory segment from which buffers are allocated for data that is read from or written to disk. Page pool size is one of the file manager startup configuration parameters.

PAL. See *Phase Alternation Line*.

part. See *object*.

patron. The term used in the Content Manager APIs for *user*.

pattern-matching character. See *wildcard character*.

PCI. See *Peripheral Component Interconnect*.

peak rate. The maximum rate encountered over a given period of time.

performance group. A group of file systems sharing system resources that can affect file system performance.

Peripheral Component Interconnect (PCI). A type of *bus* architecture.

Phase Alternation Line (PAL). The television broadcast standard for European video outside of France and the countries of the former Soviet Union.

pin. Keeping the program from being paged out after it is loaded into memory.

port. A system or network access point for data entry or exit. In the *Internet* suite of *protocols*, a specific logical connector between the *Transmission Control Protocol (TCP)* or the *User Datagram Protocol (UDP)* and a higher-level protocol or application.

port group. A logical name used to group one or more ports (network devices or interfaces) of the same network type that can be used to reach a given end-user destination. For example, if multiple *ATM* adapters in the VideoCharger Server complex are connected to the same *ATM* networks, these adapters can be configured under the same port group. The controller selects ports as necessary to balance the load.

presentation formatter. A *CGI* program that defines the forms used to select and present assets to clients.

privilege. The right to access a specific *object* in a specific way. Privileges includes rights such as creating, deleting, and selecting objects stored in the system. Privileges are assigned by the administrator.

privilege set. A collection of *privileges* for working with system components and functions. The administrator assigns privilege sets to users (user IDs) and *user groups*.

property. A characteristic of an *object* that describes the object. A property can be changed or modified. Type style is an example of a property.

protocol. The meanings of, and the sequencing rules for, requests and responses used for managing a network, transferring data, and synchronizing the states of network components.

protocol gateway. A type of *firewall* that protects computers in a business network from access by users outside that network.

proxy server. A server that receives requests intended for another server and that acts on the client's behalf (as the client's proxy) to obtain the requested service. A proxy server is often used when the client and the server are incompatible for direct connection (for example, when the client is unable to meet the security authentication requirements of the server but should be permitted some services).

purger. A function of the *resource manager* that removes *objects* from the system.

Q

QBIC. See *query by image content*.

quality of service (Do's). For an *asynchronous transfer mode (ATM)* virtual channel or a Networking BroadBand Services (NBBS) network connection, a set of communication characteristics such as end-to-end delay, jitter, and packet loss ratio.

query by image content (QBIC). A query technology that enables searches based on visual content, called features, rather than plain text. Using QBIC, you can search for objects based on their visual characteristics, such as color and texture.

R

RAID. See *Redundant Array of Independent Disks*.

README file. A file that should be viewed before the program associated with it is installed or run. A README file typically contains last-minute product information, installation information, or tips for using the product.

real time. The processing of information that returns a result so rapidly that the interaction appears to be instantaneous.

Real-Time Transport Protocol (RTP). A *protocol* that provides end-to-end network transport functions suitable for applications transmitting real-time data, such as audio, video or simulation data, over *multicast* or *unicast* network services.

rebalance. Restriping and redistributing data across the available hard disks after a disk or disks have been removed from a *file system*.

Redundant Array of Independent Disks (RAID). A collection of two or more disk drives that present the image of a single disk drive to the system. In the event of a single device failure, the data can be read or regenerated from the other disk drives in the array.

reference. Single direction, one-to-one association between a root or *child component* and another *root component*. Contrast with *link*.

remote procedure call (RPC). (1) A facility that a *client* uses to request the execution of a procedure call from a server. This facility includes a library of procedures and an external data representation. (2) A client request to a service provider located in another node.

render. To take data that is not typically image-oriented and depict or display it as an image. In Content Manager, word-processing documents can be rendered as images for display purposes.

request. The part of a Web address that follows the *protocol* and server *host name*. For example, in the address `http://www.server.com/rfoul/sched.htm`, the request is `/rfoul/sched.html`.

ReSerVation Protocol (RSVP). A resource reservation setup *protocol* designed for an integrated services *Internet*. The protocol provides receiver-initiated setup of resource reservations for *multicast* and *unicast* data flows.

Resource Interchange File Format (RIFF). Used for storing sound or graphics for playback on different types of computer equipment.

resource manager. The component of a Content Manager system that manages *objects*. These objects are referred to by *items* stored on the *library server*.

resource manager cache. The working storage area for the *resource manager*. Also called the *staging area*.

restriping. Redistributing and rebalancing data across all available and defined disks in a *multimedia file system*. This is typically done when a disk is removed from a file system for repair or when a new disk is added to a *file system*.

RIFF. See *Resource Interchange File Format*.

RLE. See *Run-Length Encoding*.

root component. The first or only level of a hierarchical *item type*, consisting of related system- and user-defined *attributes*.

RPC. See *remote procedure call*.

RSVP. See *ReSerVation Protocol*.

RTP. See *Real-Time Transport Protocol*.

Run-Length Encoding (RLE). A type of *compression* that is based on strings of repeated, adjacent characters or symbols, which are called "runs."

S

SCSI. See *small computer system interface*.

search criteria. In Content Manager, *attribute* values that are used to retrieve a stored *item*.

semantic type. The usage or rules for an *item*. Base, annotation, and note are semantic types supplied by Content Manager; users can also define their own semantic types.

server. A functional unit that provides services to one or more clients over a network. Examples include a file server, a print server, and a mail server.

Simple Network Management Protocol (SNMP). In the *Internet* suite of *protocols*, a network management protocol that is used to monitor routers and attached networks. SNMP is an application layer protocol. Information on devices managed is defined and stored in the application's *Management Information Base (MIB)*.

small computer system interface (SCSI). A standard hardware interface that enables a variety of peripheral devices to communicate with one another.

SMIT. See *System Management Interface Tool*.

SMS. See *system-managed storage*.

SNMP. See *Simple Network Management Protocol*.

staging. The process of moving a stored *object* from an offline or low-priority device back to an online or higher priority device, usually on demand of the system or on request of a user. When a user requests an object stored in permanent storage, a working copy is written to the *staging area*.

staging area. The working storage area for the *resource manager*. Also referred to as *resource manager cache*.

stand-alone system. A preconfigured Content Manager system that installs all of the components of a Content Manager system on a single personal computer.

sticky pool. The part of the *page pool* that is made available to cache the first block of frequently used interactive files. Sticky pool size is one of the file manager startup configuration parameters.

storage class. Identifies the type of media that an object is stored on. It is not directly associated with a physical location; however, it is directly associated with the *device manager*. Types of storage classes include:

- DASD
- Fixed Disk
- Optical
- Stream
- Tape
- TSM

storage group. Associates a storage system to a storage class.

storage system. A generic term for storage in the Content Manager system. See *TSM volume*, *media archiver*, and *volume*.

streamed data. Any data sent over a network connection at a specified rate. A stream can be one data type or a combination of types. Data rates, which are expressed in bits per second, vary for different types of streams and networks.

stripe group. A collection of disks that are grouped together for serving media streams. The *multimedia file system* uses stripe groups to optimize delivery of multimedia *assets*.

stripe width. The size of the block that data is split into for *striping*.

striping. Splitting data to be written into equal blocks and writing blocks simultaneously to separate disk drives. Striping maximizes performance to the disks. Reading the data back is also scheduled in parallel, with a block being read concurrently from each disk then reassembled at the host.

subclass. A *class* that is derived from another class. One or more classes might be between the class and subclass.

superclass. A *class* from which a class is derived. One or more classes might be between the class and superclass.

suspend. To remove an *object* from its *workflow* and define the suspension criteria needed to activate it. Later activating the object enables it to continue processing.

system-managed storage (SMS). The Content Manager approach to storage management. The system determines object placement, and automatically manages object backup, movement, space, and security.

System Management Interface Tool (SMIT). An interface tool of the AIX operating system for installing, maintaining, configuring, and diagnosing tasks.

T

table of contents (TOC). The list of *documents* and *folders* that are contained in a folder or *workbasket*. Search results are displayed as a folder table of contents.

Tagged Image File Format (TIFF). A file format for storing high-quality graphics.

TCP. See *Transmission Control Protocol*.

TCP/IP. See *Transmission Control Protocol/Internet Protocol*.

thin client. A client that has little or no installed software but has access to software that is managed and delivered by network servers that are attached to it. A thin client is an alternative to a full-function client such as a workstation.

throughput. A measure of the amount of information transmitted over a network in a given period of time. For example, a network's data transfer rate is usually measured in bits per second. Throughput is a measure of performance. It is also measured in *Kbps* or *Mbps*.

TIFF. See *Tagged Image File Format*.

Tivoli Storage Manager (TSM). A *client/server* product that provides storage management and data access services in a heterogeneous environment. It supports various communication methods, provides administrative facilities to manage the backup and storage of files, and provides facilities for scheduling backup operations.

TOC. See *table of contents*.

token ring. According to IEEE 802.5, network technology that controls media access by passing a token (special packet or frame) between media-attached stations.

token-ring network. A network that uses a ring topology, in which tokens are passed in a circuit from node to node. A node that is ready to send can capture the token and insert data for transmission.

topology. In communications, the physical or logical arrangement of nodes in a network, especially the relationships among nodes and the links between them.

Transmission Control Protocol (TCP). A communications *protocol* used in the *Internet* and in any network that follows the Internet Engineering Task Force (IETF) standards for internetwork protocol. TCP provides a reliable host-to-host protocol between hosts in packet-switched communications networks and in interconnected systems of such networks. It uses the *Internet Protocol (IP)* as the underlying protocol.

Transmission Control Protocol/Internet Protocol (TCP/IP). The suite of transport and application *protocols* that run over the Internet Protocol.

TSM. See *Tivoli Storage Manager*.

TSM volume. A logical area of storage that is managed by *Tivoli Storage Manager*.

U

UDP. See *User Datagram Protocol*.

uniform resource locator (URL). A sequence of characters that represent information resources on a computer or in a network such as the Internet. This

sequence of characters includes the abbreviated name of the protocol used to access the information resource and the information used by the protocol to locate the information resource. For example, in the context of the Internet, these are abbreviated names of some protocols used to access various information resources: http, ftp, gopher, telnet, and news.

user. A person who requires the services of Content Manager. This term generally refers to users of client applications, rather than the developers of applications, who use the Content Manager APIs.

User Datagram Protocol (UDP). In the *Internet* suite of *protocols*, a protocol that provides unreliable, connectionless datagram service. It enables an application program on one machine or process to send a datagram to an application program on another machine or process. UDP uses the *Internet Protocol (IP)* to deliver datagrams.

user exit. A point in an IBM-supplied program at which a user exit routine can be given control.

user exit routine. A user-written routine that receives control at predefined *user exits*.

user group. A group consisting of one or more defined individual *users*, identified by a single group name.

utility server. A Content Manager component that is used by the database utilities for scheduling purposes. You configure a utility server when you configure a *resource manager* or *library server*. There is one utility server for each resource manager and each library server.

V

video mixing. The process of dynamically inserting or combining multiple *video objects* into a single object for distribution. An example would be the mixing of commercials and broadcast programs for satellite distribution.

video object. The data file containing a program recorded for playback on a computer or television set.

video-on-demand (VOD). A service for providing consumers with movies and other programming almost immediately, per request.

video stream. The path data follows when read from the VideoCharger Server system to the display unit.

VOD. See *Video-on-demand*.

volume. A representation of an actual physical storage device or unit on which the objects in your system are stored.

W

WAIS. See *Wide Area Information Service*.

WAV. A format to store digitally recorded sound.

Web server. A server that is connected to the *Internet* and is dedicated to serving Web pages.

Wide Area Information Service (WAIS). A network information system that enables clients to search documents on the World Wide Web.

wildcard character. A special character such as an asterisk (*) or a question mark (?) that can be used to represent one or more characters. Any character or set of characters can replace a wildcard character.

workbasket. A collection of *documents* or *folders* that are either in process or waiting to be processed. A workbasket definition includes the rules that govern the presentation, status, and security of its contents.

workflow. In earlier Content Manager, a sequence of *workbaskets* through which a *document* or *folder* travels while it is being processed.

For example, claims approval would describe the process that an individual insurance claim must follow for approval.

workflow coordinator. In earlier Content Manager workflow, a user who receives notification that a *work item* in the *workflow* has not been processed in some specified time. The user is selected for a specific *user group* or upon creation of the workflow.

work item. In earlier Content Manager workflow and Enterprise Information Portal advanced workflow, any work activity that is active within a *workflow*.

worklist. A collection of *work items*, *documents*, or *folders* that are assigned to a user.

work state. The status of an individual *work item*, *document*, or *folder*.

work step. A discrete point in a *workflow* or *document routing process* through which an individual *work item*, *document*, or *folder* must pass.

World Wide Web (WWW). A network of servers that contain programs and files. Many of the files contain hypertext links to other documents available through the network.

WWW. See *World Wide Web*.

X

XML. See *Extensible Markup Language*.

Index

A

- access control list
 - moving domains 101
- accessibility 6, 123
- action 109
- ad hoc routing 111
- administrative domain 96
- administrative domains 5
- asynchronous recovery 83
 - for AIX 84
 - for Solaris Operating Environment 85
 - for Windows 84
- Asynchronous Recovery utility 81, 83
- attribute
 - text search 10
- attribute group, creating 11
- attribute, defining 10
- auto-linking
 - definition 23
 - example 34

B

- back up server data 77
- branching 109, 111

C

- cardinality 15
- cataloging 60
- child component
 - cardinality 15
 - cascade delete rule 15
 - defining 13
 - example 13
 - restrict delete rule 15
- client
 - document item types 36
 - support for data model elements 10
- collection
 - assigning to a domain 99
 - description 67
- collection point 109
 - add to a process 111
- collections
 - moving domains 100
- component 12
 - child 13
 - root 12

D

- data model scenarios
 - modeling insurance data 32
 - modeling journal article data 32
- database access password, changing 55
- db2rbind 104
- delete rule 15

- device manager
 - description 64
 - disabled 64
 - types 64
- disability 6, 123
- display name 3
- document model 17
- document part
 - classification 18
 - definition 17
 - example 34
 - ICMANNOTATION type 18
 - ICMBASE type 19
 - ICMBASESTREAM type 19
 - ICMBASETEXT type 19
 - ICMNOTELOG type 19
 - version policies 20
- document routing 104, 109
- domain 100
- domains
 - create 97
 - creating 5
 - sub administrator privileges 98
 - super administrator privileges 98
 - understanding 97

E

- event code 104
- events table
 - removing entries 104

F

- failed transactions 83
- file system, as storage system
 - description 65
- foreign keys
 - advantages and restrictions 22
 - definition 25
 - example 25, 34

G

- grant privilege set 96

H

- hierarchical item types 12, 13, 15
 - child component 13
 - root component 12

I

- icmprepenv.bat 84
- icmprepenv.sh 84
- icmrmdel.bat 84
- icmrmdel.sh 84, 85
- icmrmtdel.bat 84

- icmrmtdel.sh 84, 85
- ICMSTITEMEVENTS 104
- ICMSTSYSADMEVENTS 104
- index class 13
- insurance scenario 34
- item 104
 - classifying as an item type 16
 - definition 21
 - version policies 19
- item type 15
 - classifications 16
 - defining 15
 - example 15, 34
 - subset 20
 - view 20
- item type classification
 - document 17
 - document part 18
 - item 16
 - resource item 16

J

- journal article scenario 32

K

- key field 10
- keyboard 6, 123

L

- lan cache 70
- language codes 55
- LDAP
 - configuring 93
 - importing 93
- library server
 - backing up data 77
 - configuration profile 53
 - configuring 53
 - event table log 115
- library server monitor fail-over service 68
- links
 - advantages and restrictions 22
 - auto-linking 23
 - definition 22
 - example 22, 34
 - link types 22

M

- media manager 61
- media object class
 - predefined types 28
- media object class, definition 28
- media server 61

- migration
 - change date 106
 - remote 106
 - schedule 105
- migration policy 106
 - create 105
 - description 66
 - remote migration 105
- migrator schedule 106
- MIME type, definition 26
- modeling data
 - deciding whether you require a
 - custom data model 45
 - diagramming your data
 - relationships 45
 - identifying hierarchies and elements
 - that might have multiple values 43
 - identifying the elements that might be
 - searched for 42
 - identifying your data 37
 - identifying your users and what data
 - they need to access 41
 - in Content Manager 46
 - separating your data into operational
 - and non-operational data 39
 - sorting your data into like types 40
- multi-valued attributes 13

O

- object storage
 - overview 63
- object, definition 26

P

- privilege group 95
- privilege set 91, 95
 - creating 95
 - moving domains 101
- procedures, system administration
 - backing up data 77
- process
 - define 109

R

- rebind 104
- references
 - advantages and restrictions 22
 - definition 24
 - example 24, 34
- remote migration 105, 106
- reorgchk 103
- replication 67
- resource manager
 - assigning to a domain 99
 - assigning users to 96
 - changing password to 54
- resource manager services 82
- resource manager utilities 81
- resource manager, moving domains 100
- restore data consistency between
 - servers 83
- restore server data 77

- root component
 - defining 12
 - example 12

S

- Secure Sockets Layer 58
- selection 109
- semantic types
 - definition 21
 - predefined semantic types 21
- server
 - synchronize 77
- Server definitions, creating 67
- servers
 - backing up data 77
 - configuration profile 53
 - configuring 53
 - restoring data 77
- setprocnv.bat 81
- setprocnv.sh 81, 82
- single sign-on 54
- SSL 58
- staging area 61
- storage class
 - description 64
- storage group
 - description 66
- storage management
 - collection 67
 - device manager 64
 - migration policy 66
 - storage class 64
 - storage group 66
 - storage system 65
- storage system
 - assigned 65
 - description 65
 - overflow 65
 - unassigned 65
- synchronizing servers 77
- system administration client 5
 - logging on 5
- system administration procedures
 - backing up data 77

T

- tables
 - reorganize 103
- text search 29
 - of attributes 30
 - of documents 30
 - of resource items 30
- text search options 30
- translation 55
- TSM, as storage system
 - description 65

U

- user 91
 - moving domains 99
 - privilege set 96
- user group 96
 - moving domains 100

- user ID 91

V

- validation utilities 81, 85
- version policies 19
- Video Charger 61

W

- work basket 109
 - add to a process 111
 - define 110
- work node 109
- work package 109



Program Number: 5724-B19

Printed in U.S.A.

SC27-1335-01

