

# Table of Contents

	Title	Page No.
1.	Introduction	2
2.	Why Content Manager for e-Gov?	4
3.	Architecture of Content Manager	7
4.	Understanding the terminology	10
5.	Access Control	14
6.	The Folder Manager Data Model	17
7.	The Storage System	20
8.	Developing a client application for Content Manager	25
	<a href="#">Appendices</a>	
A	Installing Content Manager	29
B	A Brief Discussion of features of CM	32
C	Migration policies and Logging in Content Manager	35
D	Additional Products for Content Management from IBM	38
E	Useful References for Content Manager	40

# Chapter 1

## Introduction

### What is Content Manager?

Simply put, Content Manager is a document management application, providing many sophisticated features and easily adaptable to a network solution for data access.

It can help you maximize the value of your information and multimedia assets. Regardless of the type of data (text documents, scanned images, audio, video, forms any binary object), it lets you store data on distributed servers and access it through a single point without knowing about the whereabouts of the data on the network.

This is ideal, where large amounts of data of heterogeneous nature is required to be accessed through a single client application where the data could actually be located across various servers.

**For example** a library with various branches across the country may place its books in popular formats e.g. pdf format and other media such as images, videos etc. across various servers which can be accessed through a central server using the Content Manager solution. Any user can then have a single view of the entire library using a web-front end.

It lets you...

- ◆ Enable e-business applications with business content— documents, images, computer output, forms and multimedia
- ◆ Automate work processes to ensure best practices throughout your enterprise
- ◆ Enhance customer satisfaction with content-driven Web self-service
- ◆ Provide an enterprise wide view of all-content related to your customer relationships
- ◆ Differentiate your Web presence by delivering high quality multimedia content

**Fig 1: Features of Content Manager**

IBM Content Manager capabilities include :

- ◆ Document imaging.
- ◆ Document management.
- ◆ Rich media management.
- ◆ Computer Output and Enterprise Report Management.
- ◆ Archiving in SAP® and Lotus® Notes® environments.
- ◆ Streaming audio and video.
- ◆ Enterprise wide search from Internet or intranet clients.
- ◆ Automated workflow.

**Fig.2 : Capabilities of Content Manager**

Another description of Content Manager from an IBM Brochure:

“IBM Content Manager – a comprehensive set of Web-enabled, integrated software solutions for content management –makes all of your enterprise information readily accessible to anyone, from anywhere. Empowered by immediate and seamless access to digitized and indexed content, authorized users throughout and beyond your organization can maximize efficiencies by using information, rather than wasting time searching for it.”

## Chapter 2

# Why Content Manager for e-Gov?

The following points discuss how Content Manager differs from other existing data management solutions.

1. Content Manager brings the colossal volumes of information businesses produce within easy reach of users and customers in the office and on the Web. But you already have access to that data through your database, right? Well, considering that only 15 percent of most corporate information resides in structured formats such as relational databases, access to the other 85 percent isn't always that easy. And, when you consider the amount of information that exists in the business world today, the problem becomes even more complicated.

2. The problem with having all this information online is being able to find and assemble all the bits and pieces needed for a given task. For example, before recommending a job candidate for a position, a recruiter needs all the candidate's information, including a resume, interview (audio or video), passport, email correspondences, and official transcripts, plus the details of job vacancies in the company. To be most effective, this information must be available electronically from a single source. Content Manager lets you develop applications that pull together information, regardless of format, and organize end-user presentation in a meaningful way

3. The motivations for upgrading from analog to digital are obvious: speed, accuracy, improved search capability, workflow automation, and asset protection. Take the example of a mortgage provider's records on an individual borrower. A non-digital process might involve a manila folder holding originals or copies of the loan application, faxes, various supporting documents, and so on. This folder would pass manually from one worker to the next, with each worker making copies and upgrading the application status. But if you have a digital version of all that information, workers can use Content Manager to store and interact with an electronic version of the folder. That folder could include everything the manila folder contained, plus audio recordings of phone conversations (remember the recorded message that alerts customers that they may be recorded for quality purposes?), email messages, scanned documents, text documents, and other electronic information. In essence, Content Manager provides the same kind of complete information management for this kind of electronic information that traditional relational databases provide for structured data.

4. A single, integrated product, Content Manager now handles the rich media and document management functions the earlier products provided. And, through add-on components, Content Manager also manages streaming video and computer-generated output and provides repositories for information from SAP and by email from Lotus Domino.

5. The architecture of CM allows object servers to be remotely located near user groups that create and maintain the content. **And each object server can reside on a different operating system.**

6. End users generally fall into one of three categories, including:

- Scanning clients, who feed content into the system
- Indexing clients, who create content metadata
- Display clients, who search on, view, and update content.

The first group will scan or acquire the content, populate simple index keywords (such as content source), annotate it, and forward it to the indexing clients. The indexing clients, who spend time considering the significance of the content and how it will be used, add more context-specific keyword information for grouping related content into document folders. For example, a video of a keynote address from the company president may touch on many subjects. Index clients may need to add hundreds of keywords as content metadata so that the video shows up correctly in search results. Display clients are those users who access and update the content entered and annotated by the other clients.

7. To manage workflow, Content Manager lets you put workers and work processes on a workflow path. Content folders containing related documents are passed as objects from one work node to the next along the workflow path. Each worker has a workbasket in which folders are dropped for review, annotation, or promotion to the next node on a workflow path.

8. The user can perform structured, unstructured, parametric, or contextual content queries. ?

9. The system administrator can define archiving criteria to specify how long objects will reside on one medium before being migrated to another. For example, the administrator can specify criteria that make closed accounts not referenced for one year qualify for migration from hard disk storage to optical disk.

10. Content Manager **EIP Client Kit** provides Web access to Content Manager data through IBM Enterprise Information Portal, an architecture and middleware product that enables direct access to intranet and Internet sources. Local

applications, Web sites, and queries tailored from disparate sources are all placed on a user's portal front page for quick access.

11. Both DB2 Extenders and Data Links Manager enable users to unite information under DB2's control, providing powerful information integration at the repository level. (Content Manager, in contrast, provides powerful integration across varying repository types.) Each technology works toward unification, integration and simplified access.

# Chapter 3

## Architecture of Content Manager

The Content Manager architecture consists of a Client, Library Server and Multiple Object Servers.

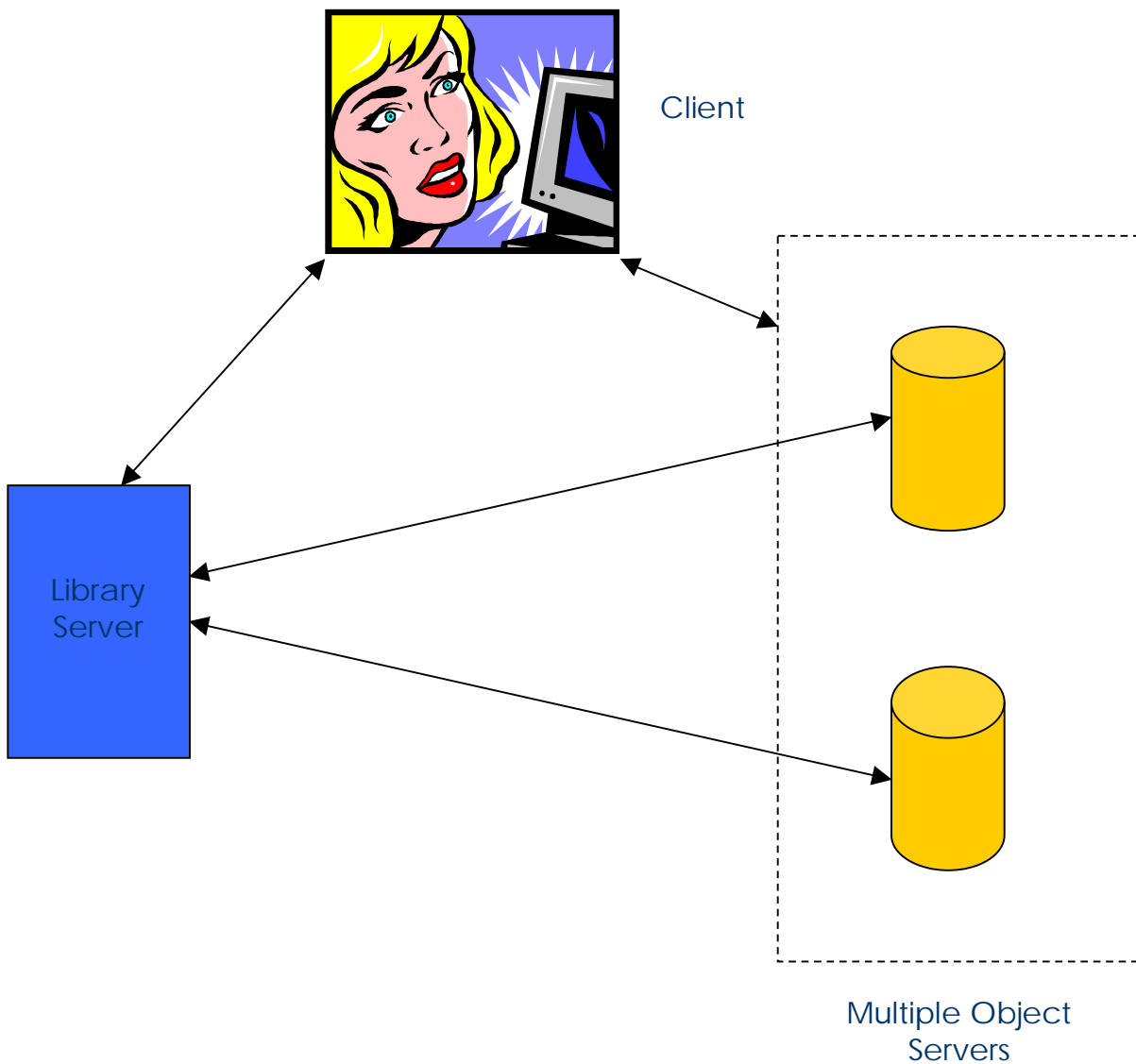


Fig: Architecture of Content Manager for a single Implementation

The above figure shows the architecture for a typical Content Manager implementation. It consists of:

## Client

This is the end user who will be accessing the data stored in the Content Manager implementation using Library Servers and Object Servers. Note that the client communicates with both Library Server as well as the Object Servers. The connection between the client and the object servers has high bandwidth as compared to the connection between the client and the Library Server. This will become clear in the following sections.

## Library Server

Currently, there is one Library Server per implementation. By an implementation, we mean an independent implementation of a Content Manager solution.

### **It's Purpose**

Library server, manages the Content Manager catalog information, locates stored objects using a variety of search technologies, provides secure access to the objects in the collection, and communicates with the object servers. A Content Manager system requires one library server, which can run on Windows NT<sup>(R)</sup>, Windows 2000, AIX, or OS/390<sup>(R)</sup>.

The library server uses a relational database to manage digital objects and provide data integrity by maintaining index information and controlling access to objects that are stored on the object servers. This relational database can be IBM DB2 Universal Database<sup>(TM)</sup> (DB2 UDB) or Oracle.

### **An Analogy**

Consider a library near you. When you enter the library, there are various floors with lots of racks containing thousands of books on various topics and you have no idea whereto go if you want a book say "War and Peace" by Leo Tolstoy. So first you need to go to the Catalog Section of the Library, which hopefully indexes all the books in the library based on the author as well as the title. You look up the title and get the Rack No, which contains the above-mentioned book. So, here you are the client, the Catalog Section acts as the Library Server and the Racks as the multiple Object Servers.

You have no idea which Object server to go to, but with the help of the Library Server, your client can retrieve the data from the appropriate Object Server.



In addition to cataloging data on multiple attributes, the Library Server also provides other essential services such as secure access to data , user authentication and integrity of data stored on the Object Servers.

## Object server

The object server is the repository for objects that are stored in the Content Manager system. Users store and retrieve objects from the object server, through requests that are routed by the library server. . There can be multiple Object Servers in an implementation. The object server efficiently and automatically manages storage resources, based on the storage management entities (such as volumes) that are defined using the Content Manager system administration program A Content Manager system can have many object servers distributed across networks to provide convenient user access. Object servers run on AIX, Windows NT, Windows 2000, or OS/390.

A database on the object server contains data about the exact location of each object. The database can be either DB2<sup>(R)</sup> UDB or Oracle.

## Advantages

The above architecture provides the following advantages:

- Support for multiple, distributed object servers allows you to store digital objects close to the users who need to access them frequently. This support is especially important for delivery of large multimedia objects.
- Support for heterogeneous servers allows you to use the system for all kinds of data (including streamed data), while optimizing the processing of individual data types.
- Client communication through the library server ensures the integrity of data objects. A client can access objects only by requesting them through the library server. The library server database contains information such as object types, indexes of all stored objects, authorized users of the system, and access control lists for each object.
- Separation of client applications, indexes, and data makes applications independent of the data's location on the servers.
- The open architecture allows the intermixing of additional object servers on the same or different operating systems and supports scaling from one operating environment to another, as growth requires after implementation.

# Chapter 4

## Understanding the terminology

The Content Manager client interface be divided into two layers:



### What is the difference between the two layers?

The Library Client Layer is the lowest layer which using which the data can be accessed stored in the Content Manager implementation and the system can also be configured like access control, storage policies etc.

Folder Manager is a logical data model built over the lower layer, which can be adequately used to store and access data. This model provides many convenient features, which can be used for document management instead of directly using the Library Client API, which can be more difficult in nature. The Folder Manager Logical Model may suffice for most needs.

Note that one can just the Library Client API if he so desires, but Folder Manager API can be easier and more convenient to use in most circumstances. But for some tasks like configuration, Library Client API must be used. Folder Manager API only provides for managing data on the object servers.

Let us understand the following concepts:

- q Items
- q Parts
- q Index Classes
- q Access Control

### Notion of Content Class

Content Class essentially means the data format of the particular object. For e.g. BMP format is a separate Content Class and PDF is another. You can replace Content Class by Data Format while reading. E.g. A Picture may belong to the BMP content class or the GIF Content class.

## Items and Parts

Items are the basic building blocks in Content Manager. Basically an item consists of a set of related data, which can be treated as an independent entity. By independent entity we mean that features like security control can be applied to each item separately. An item consists of various parts, which make up the complete item. Each part is a binary object or BLOB (Binary Large Objects). Each part belongs to a particular content class and within an item; different parts belonging to different content classes can be stored.

Each item can be assigned various attributes, which fall into two categories:

- q **System Attributes** - an item identifier and a creation timestamp
- q **User Attributes** - The item identifier allows the user to assign user-defined attributes to an item. You can define various key words, which describe the item. User-defined attributes must include an **index class**. An **index class** categorizes and stores items according to their attributes. User-defined attributes can also include properties, links and names

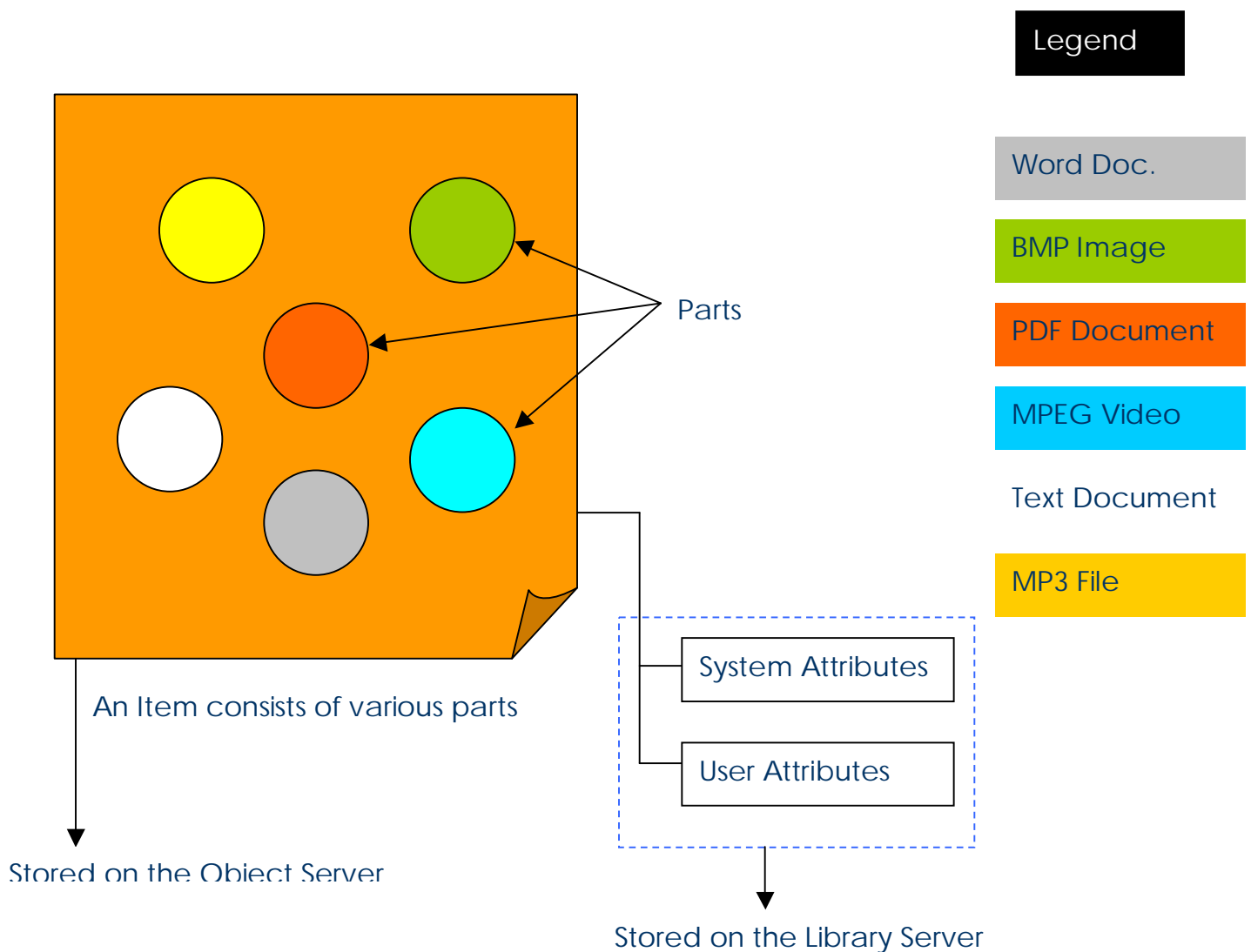


Fig: An overview of an item

# Understanding Index Classes

## Overview

We will look at

- q What are Index Classes and attributes?
- q Derived Index Classes
- q Index Class Views
- q Properties

## What are Index Classes and attributes ?

Each item you store in Content Manager will belong to a definite type which will be further defined by some **attributes**. For example some items may belong to the License Type, some may belong to Passport Type etc.

**Index Class** refers to the type of the item being stored in the Object Server. Each item must belong to a particular index class to define its “type”. Each **index class** will further consist of **attributes** which give a brief description of the item being stored under that category. For example , License Index Class may contain the attributes : Name , Age , Address , Date of Issue etc.

It is important to understand that while querying for data in Content Manager , you can search over these attributes to locate a particular item. So you should define all the searchable properties of a particular Index Class as attributes o/w a search cannot be conducted over that property. For Example , In the Licence Index Class you may only define Name and Date of Issue as attributes if you don't need to search using other properties of the item like Address etc.

**Note** : In Content Manager Version 7.1 multiple value attributes are also supported.

## Derived Index Classes

This concept is similar to Inheritance in Object Oriented Programming. Using a previously existing Index Class , you can define a new index class which adds some more attributes in addition to the existing index class.

## Index Class Views

This is used to restrict access into a particular index class. For example , you may define an Index Class containing 20 different attributes. However , you want that some people have access only to the first 10 attributes and they can't view or search over the other 10 attributes. Then you can define an “Index Class View” which is an incomplete window to the Underlying Index Class and this limited view can be used to search over the attributes over which it provides access. You can have multiple views for multiple users , thus obviating the need of multiple index classes.

Note that all access operations and query operations are defined only over index class views and not on index classes. When you define an index class, a default view is also created containing all the attributes. You can further define more restrictive view for finer control.

## Properties

In addition to attributes, there may be some ad-hoc properties of the item which need not be searched and indexed. You can attach “type=value” tags to the item to define its properties.

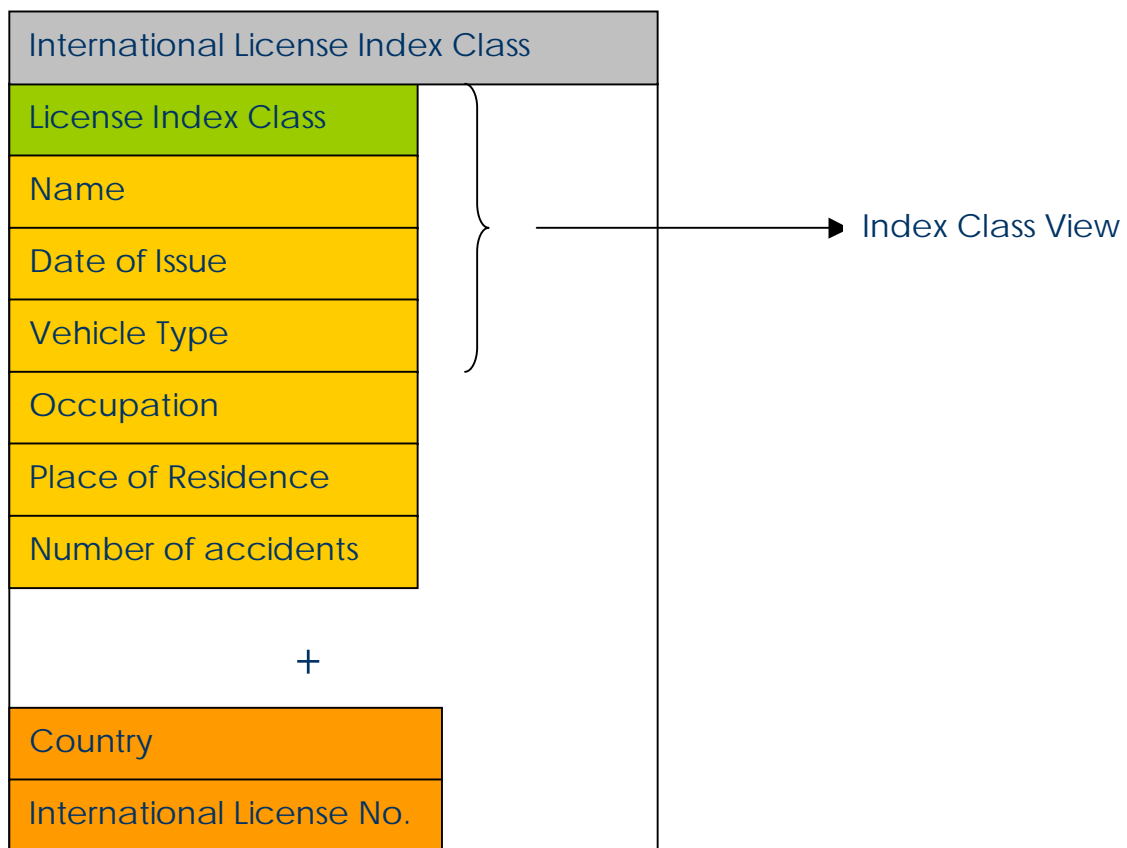


Fig: Example of a Derived Index Class and an Index Class View

# Chapter 5

## Access Control

Content Manager features a two-way security system for accessing the stored data.

1. First, the user must have the privilege to execute that particular operation such as delete. This step is general and not in reference to a particular item.
2. The item which the user is attempting an operation upon, must allow that particular user to perform that particular operation.

Thus, even if a user has the privilege to delete defined in its general privileges, it cannot delete an item unless the item allows that particular user to delete it and vice versa. Thus both conditions are required to be true for the operation to be successful.

### Overview of Access Control

- ◆ Privilege Strings
- ◆ General Privileges
- ◆ Users
- ◆ User Groups
- ◆ Access Control Lists

### Privilege Strings

A Privilege String is a **set** of privileges such as add, update ,delete and various others operations defined in Content Manager. A Privilege String is used to grant privileges to users or groups.

### Users

One must add users to the Content Manager to enable them to access the stored data or to configure the system. When adding the user, one must also define the password and the privileges the user has while accessing the data and the Content Manager implementation in general. These privileges defined for the user are known as **General Privileges**. Any operation which a user requests must be allowed in the **General Privileges** which have been granted to the user. Note that the **General Privileges** consist of a privilege string, which defines all the operations, which are allowed.

## User Groups

A user group is a named collection of individual users of the library. You define groups to control access to information at the group level and to simplify administration of access control. For example, you can create a group and have all users be members of the group.

Instead of managing individual users and their rights to access individual items, it is easier to create groups and then map the group rights to the items. One can then easily add or delete users from the group.

This will become clearer after understanding Access Control Lists

## Access Control Lists

An Access Control Lists consists of a list of users and their associated rights. The ACLs are associated with items or index classes to take care for other side of the access control mechanism, to control access from the item's side.

### An Example

#### Access list table with two-access control lists

AccessListID	User	UserKind	Privilege
ACL_1	Barbara	User	Priv_1
ACL_2	All users	Public	Priv_21
ACL_2	Sam	User	Priv_22
ACL_2	SalesGroup	Group	Priv_23

The above figure shows a list consisting of two access control Lists. ACL\_2 has 3 entries in its list. It states that all users have privilege Priv\_21 on the item with which this ACL is associated. Similarly, Sam has the right to privilege Priv\_22 and SalesGroup group has privilege Priv\_23 on the items associated with this ACL.

#### Advantages of using User Groups over Individual Users

UserKind consists of three types: Public, User, and Group. Public refers to ALL users, User refers to an individual users and Group refers to a group.

One may note that why it is easier to create groups than create individual users because if there are 1000 users then one will have to add the individual users in the ACLs associated with items to define their rights for these items and one may have a 1000 x 1000 mapping if there are 1000 items in the library. However, one

may divide the user into groups and then just add groups to ACLs for easier management of ACLs.

For further details, please refer to the documentation that comes with Content Manager. There are additional concepts like Proxy users, Owner Enabling and Access Control List identifiers, User Exits.

## Setting up access control

Before users can submit requests to the library, you must set up access control components so that the access control algorithm can decide whether to grant requests. To set up access control, complete the following steps:

1. Define privilege strings that you want to associate with users.
2. Define all users of the library.
3. Define user groups.
4. Create items that act as ACL identifiers. Content Manager requires that all ACLs be valid objects in the library.
5. Define user privileges for ACLs. Access control lists are bound with items and index class views. This step identifies user privileges for items and index class views.

An item or index class view can have a defined ACL identifier associated with it. Until ACLs are associated with the ACL identifier, no one can use the item.

After you complete these steps, the system binds every item and index class view that are created in the library with an ACL.

## The Access Control Algorithm

This algorithm determines the steps which are taken to ensure the validity of an operation request on an item by a particular user.

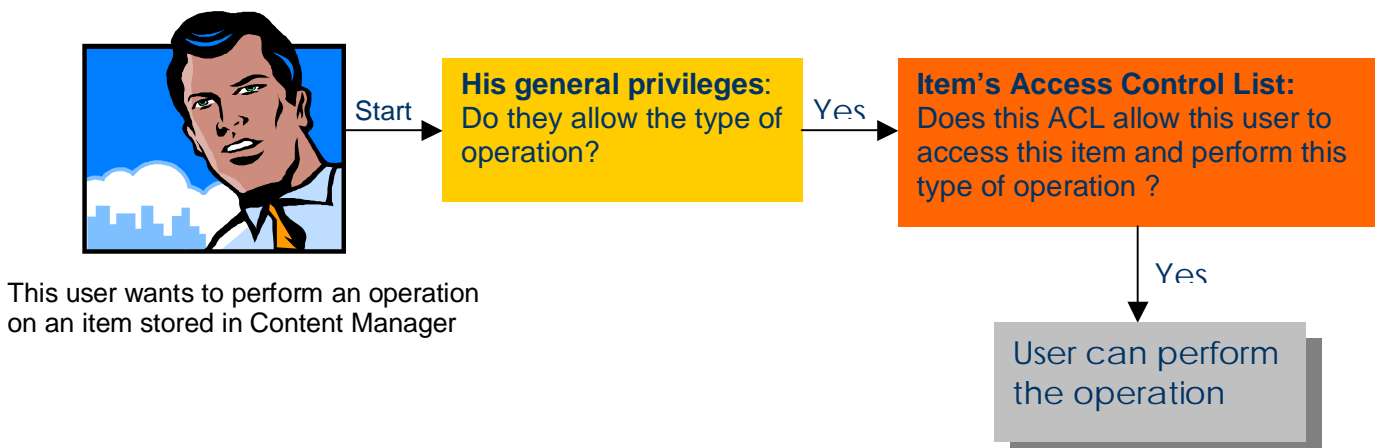


Fig: Checks performed for a user operation over an item



# Chapter 6

## The Folder Manager Data Model

The folder manager implements a logical data model that provides your application with many capabilities for managing business objects. The folder manager is one of the data models you can use with the Content Manager library.

To simplify the explanation of this data model, the term "document" is used to mean any multimedia, digital object. The above figure provides a high-level view of the logical data model for the folder manager. In the folder manager data model, the basic components are documents, folders, workbaskets, and workflows.

Documents in the folder manager are similar to paper documents. A document consists of a set of closely related objects, such as pages in a letter or report. Documents can contain one or more **parts**. These parts, known as **base parts**, can be pages or illustrations in a letter, report, or other documents. Other parts associated with documents are **annotations, notes, and events**.

### **Annotations**

An annotation part associated with a document can highlight sections of a document and display sticky notes, stamps, or freehand drawings.

### **Notes**

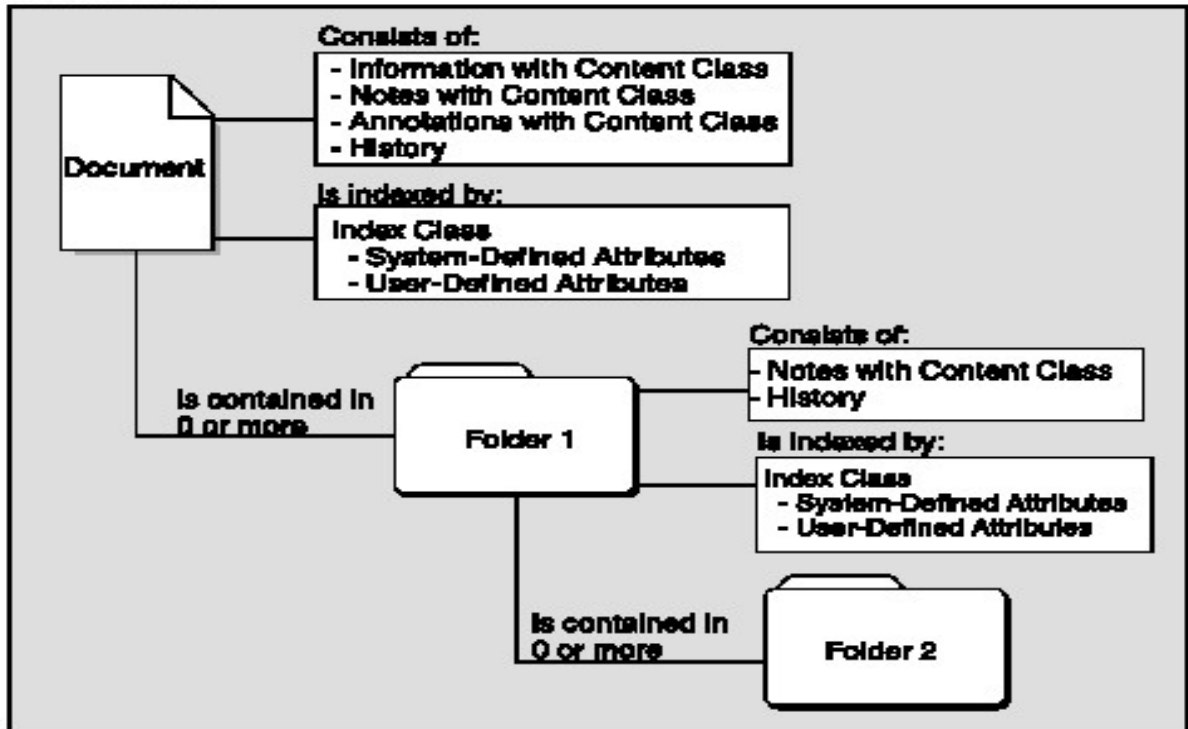
A note part associated with a document is textual information that you attach to the document to give additional information to other users. For example, you might attach a note to draw the reader's attention to part of the document.

### **Events**

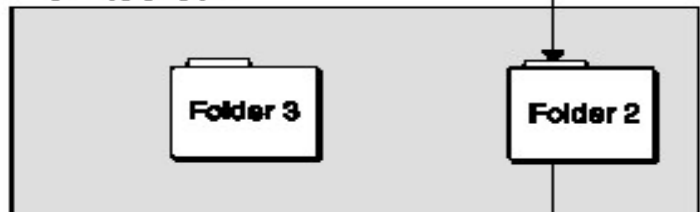
An event part associated with a document provides a historical trail of the processing you performed on the document. You can associate only one event part with a document.

Folders in the folder manager are similar to folders in a paper filing system. Each folder can contain one or more documents or other folders. Each folder has a table of contents that lists all the documents and folders it contains. You can associate note parts and an event part with a folder. You can place a folder in a workbasket as an item to be worked on and assign a folder to a workflow process.

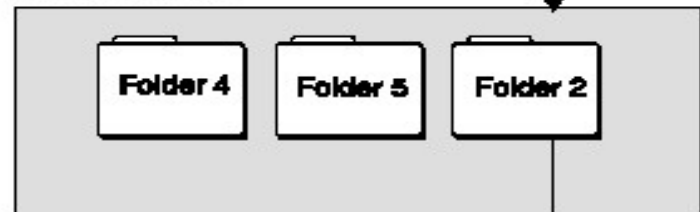
### Workbook 1



### Workbook 2



### Workbook 3



Output of this workflow

Fig: The Folder Manager Data Model

## Workflow in Content Manager

The folder manager provides a number of functions that let your applications create simple business processing models and then route the work through the process until it completes. Your application controls how you process the work. The folder manager does not provide automatic routing facilities.

The folder manager uses workbaskets and workflows. A **workbasket** is a container that holds documents and folders that you want to process. A **workflow** is an ordered set of workbaskets that represent a business process. Folders and documents move from one workbasket to another within a **workflow**. When you place a folder or a document in a workflow, it is a work-in-process item.

You must create workbaskets and workflows using the Content Manager system administration program before you can use folder manager functions to route documents and folders. When you create a workbasket, you can specify characteristics for it. For example, when items from the workbasket are requested for processing, you can indicate whether the system dispatches the highest priority document or the user is able to choose items for processing. You can also indicate a trigger limit that, when reached, allows your application decide whether or not to add more items to the workbasket.

When you create a workflow, you can also specify characteristics. For example, you can specify whether the system should keep or discard the history records for an item that has completed processing in a specific workflow.

In general, items (folders or documents) move from one workbasket to another within the workflow until the process completes. As the items move within a workflow, the folder manager creates a history log for them. The log indicates the date and time that the item is moved and the user who moved the item. When the item completes processing, the system either discards or stores the history as an affiliated object based on the criteria set for the workflow.

# Chapter 7

## The Storage System

The storage system in Content Manager is slightly complex and involves new terminology used in Content Manger.

### Overview

- ◆ Device Managers
- ◆ Storage Classes
- ◆ Storage Systems
- ◆ Storage Groups
- ◆ Collections

### Device manager

A device manager is software that acts as an intermediary between your library server machine and physical storage.

You assign device managers to a **storage class** so that the **storage class** can communicate with the **storage systems** defined with it in a **migration policy**.

### Storage class

A storage class is a logical grouping of similar storage types (E.g. Magnetic storage , optical storage ). It requires one or more **device managers**, which you must create first.

The storage class determines the type of media that an object is stored on in each step of the object's migration. For every storage class that is used in the **migration policy** for a **collection**, there must be at least one **storage system** that is associated with that **storage class** in the **storage group** for the **collection**.

Before you create a **storage class**, you must decide which type of **storage systems** will be associated with the **storage class** so that you can install the proper device drivers.

**Example:** You want all video clips to be stored on a hard drive for 39 days before moving them to an optical drive permanently.

You create one **storage class** called *harddrv* and assign a *hard drive device manager* to it. You create a second **storage class** called *optical* and assign an *optical drive device manager* to it. You would then use both these **storage classes** to create your video clip **migration policy**.

Note:

You must create any **device manager** that you want to assign to the storage class before you create the storage class.

Remote and local storage classes are defined according to the **migration policy** in which you define them. A local storage class points to storage on the same object server as the **migration policy**. A remote storage class points to storage on a different object server than the **migration policy**.

## Storage system

A storage system is the location where an object is stored. There are different types of storage systems in Content Manager:

- Volumes, on Windows
- File systems, on AIX
- TSM, on AIX and Windows
- Media archives, on AIX and Windows

For Windows, a volume is defined as a physical or logical partition on a hard disk drive.

### Migration

*An object migrates from one storage system to another storage system based on the storage class that is assigned to the storage system.*

Note:

You must create the storage class that you want to assign to the volume before creating the volume.

Volumes must be attached to the same machine as the object server.

When you change a volume, you cannot change the storage class if there is data stored in the volume.

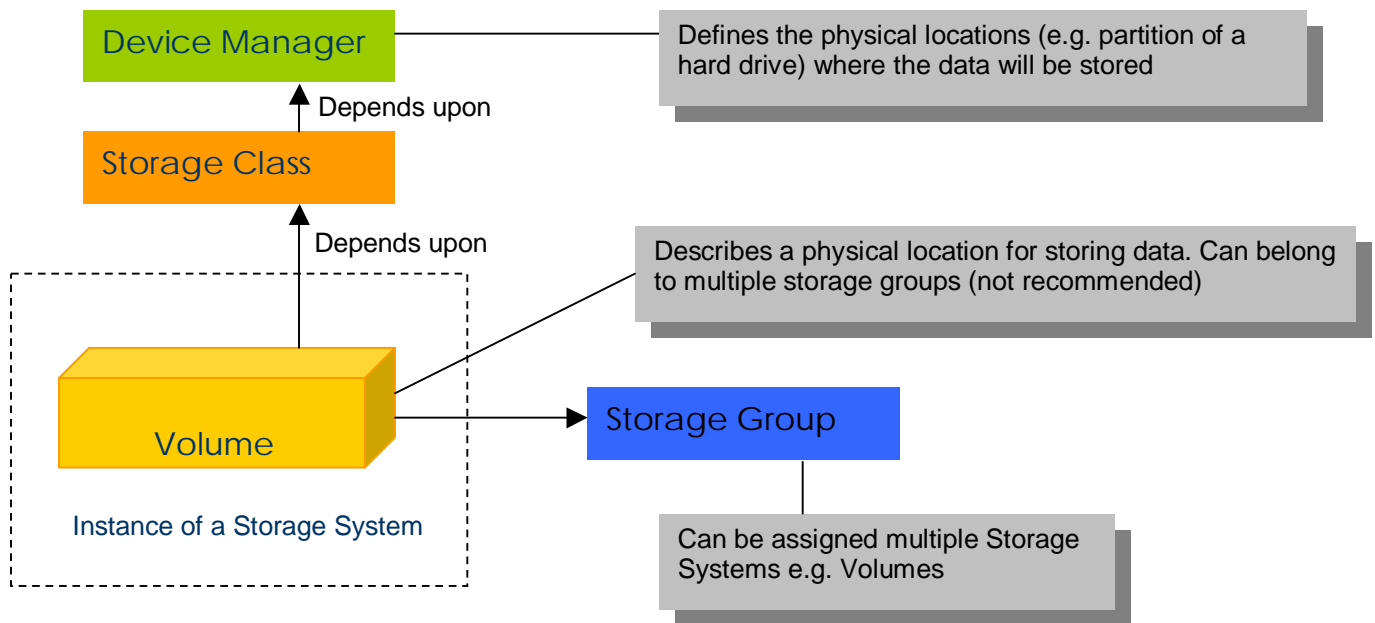


Figure: The relationships between the different elements of storage in Content Manager

## Storage group

A storage group is a logical collection of **storage systems**. You can create storage groups to limit the number of **storage systems** that an object can use for storage.

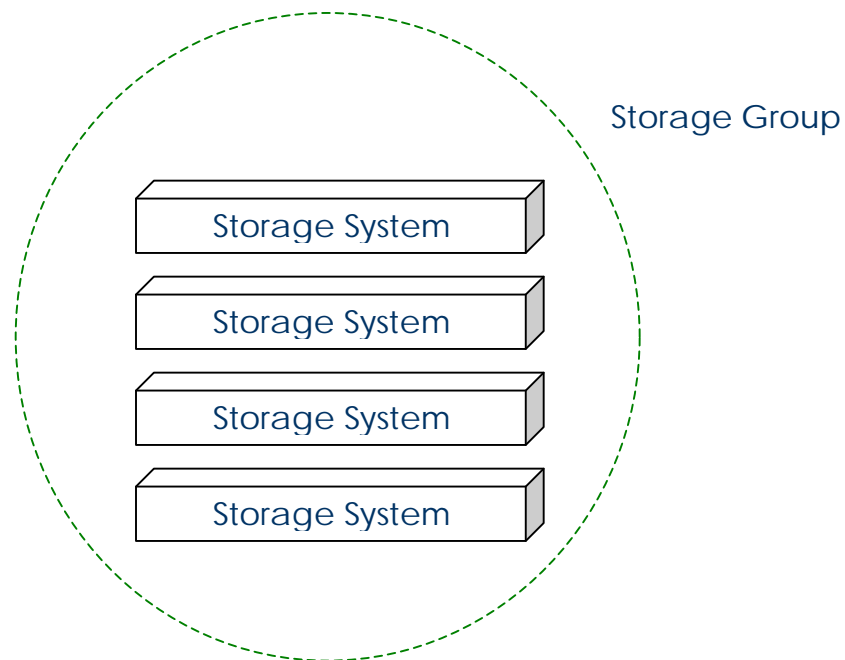


Fig: Storage Group

**Example:** You have two types of documents: articles and forms. You want both types of documents to use the same migration policy because you want them to be stored on a hard drive for two weeks and then moved to a TSM archive. However, you do not want both the articles and the forms on the same hard drives. You create two storage groups, one for articles and one for forms. You associate the articles storage group with drives E and F. You associate the forms storage group with drive G. When the migrator moves an article to the hard drive storage class, the migrator will go only to drive E or F, because the articles storage group points the migrator to those drives.

### **Migration:**

Each **storage system** in a storage group is associated with a **storage class**. *When an object in a collection is migrated to the next **storage class** in the migration policy, it is moved to a **storage system** that is associated with the target **storage class** and in the storage group that is assigned to the collection.*

### **Recommended way of storing data**

Use the system administration program to create the storage groups that you need to fit the needs of your business. ***You can assign the same storage system to more than one storage group, and you can use the same storage group for more than one collection.*** However, you might prefer not to store objects from different **collections** on the same physical devices. In that case, use different **storage systems** for each storage group and different storage groups for each **collection**.

## Collection

A collection is a group of related objects that are stored in the same storage group and are managed by the same migration policy. Objects in a collection can be replicated or copied to a copy object server for enhanced availability and security.

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

The collection provides a way to group related objects. All objects in the collection are stored on the **storage systems** that are part of the **storage group** that is assigned to the collection. All objects in the collection migrate according to the rules that are defined for the **migration policy** that is assigned to the **collection**.

### **Notes**

You must create the **storage group** and **migration policy** that you want to use in the **collection** before you create the **collection**.

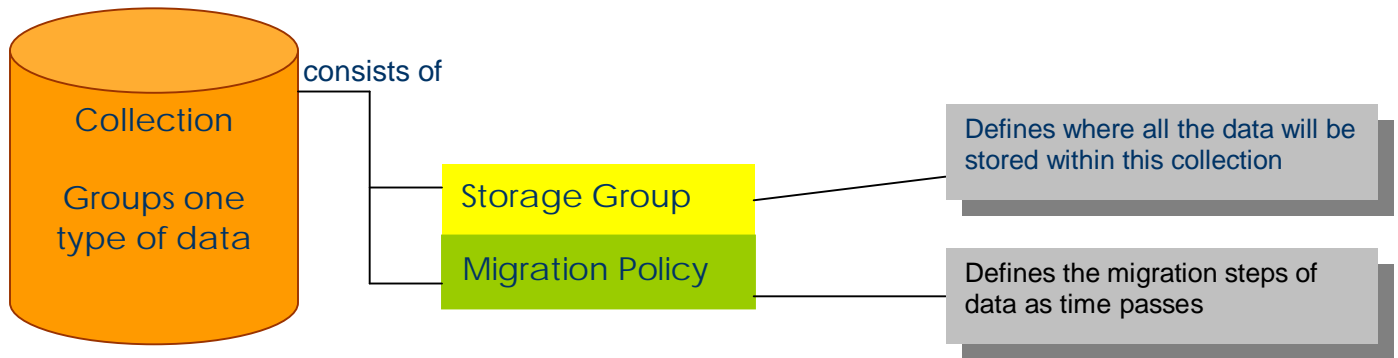


Fig: A Collection

## Migration policy

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.

The migration policy defines how long an object stays in a location and where the object will be moved to 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. You must associate the **storage class** with at least one **storage system** in the **storage group** for the **collection** to which the object belongs.

To migrate an object to another object server, specify a **remote storage class** as the final step in a **migration policy**. Use the system administration program to create the **migration policies** that you need to fit the needs of your business. You can use the same **migration policy** for more than one collection.



# Chapter 8

## Developing a client application for Content Manager

Content Manager comes with a C++ ( and JAVA )API to develop Custom Clients for Content Manager. Documentation is currently available for C++ Application Development.

There are two sets of APIs:

Library Client API and Folder Manager API. Any API or both API together can be used. If the Folder Manager data model is sufficient for your needs, you can stick to the Folder Manager API for developing your client application.

Please Note:

Before developing an application make sure the environment variables for PATH, LIB and INCLUDE which are changed by the DB2 and Content Manager installation have been copied into the System Variables list. See Installing Content Manager for details.

Steps required for developing an application in Visual C++ 6

### **Getting started**

To write Windows-based applications using Content Manager APIs, you must install the Content Manager APIs for Windows. During the installation process, the installation program will automatically update your AUTOEXEC.BAT file with required changes or Windows Registry. AUTOEXEC.BAT is only available on Windows 95 and Windows 98.

The installation program also installs Content Manager header files. These header files provide the API prototypes and data structures to use when writing a Windows-based client application for Content Manager. By default, the installation program places the header files in the \INCLUDE subdirectory of the install root directory.

Your application must include the Content Manager header file FRNP.H. The header files contain structures and data types required by Content Manager

### **Programming with the Folder Manager API**

Following steps are required:

## Include files

Your application can include the folder manager header files in two ways. You can either define FRN\_INCL\_FI and FRN\_INCL\_FO before including FRNP.H and FRNPCAPI.H header files, for example:

```
#define FRN_INCL_FI /* Include frnpfi.h & frnpfi2.h */
#define FRN_INCL_FO /* Include frnpfo.h */
#include "FRNP.H
```

Or, you can include the folder manager header files specifically, such as the following:

```
#include "FRNP.H"
#include "FRNPFI.H"
#include "FRNPFI2.H"
#include "FRNPFO.H"
```

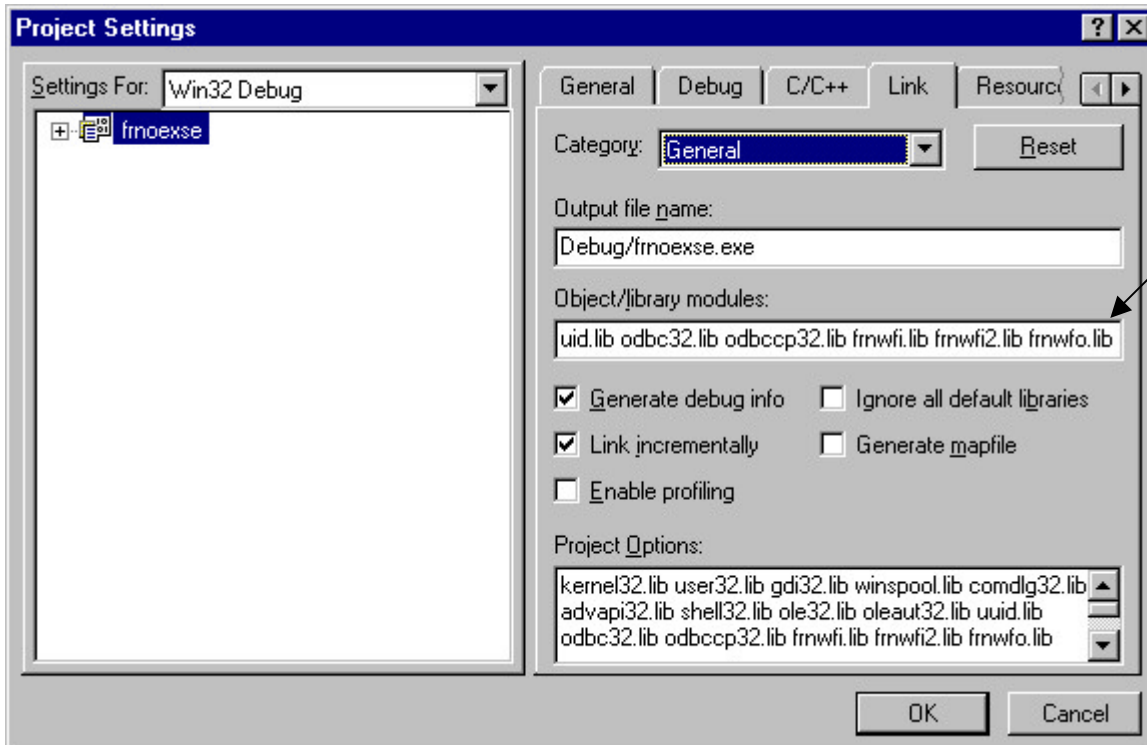
## Lib files

The folder manager lib files that your application must link to are:

- FRNWFI.LIB
- FRNWFI2.LIB
- FRNWFO.LIB

Steps:

To include the above library files, make sure the C or C++ file is a part of project workspace. If it is not, create a default workspace by compiling it (A default workspace will be created even if it fails to compile). After that go to Menu -> Project -> Settings... -> Link and in the field labeled **Object/library modules** enter the name of the above four files in the list. See the figure below:



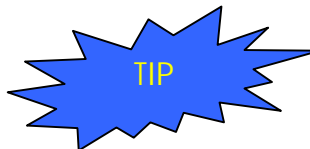
## Programming with Microsoft Visual C++ Version 6

To compile an MSVC program that calls Content Manager APIs, set the compile option FRN\_WINNT.

Steps:

To set the above option go to Menu -> Project -> Settings... -> C/C++ and in the field labeled **Preprocessor definitions** append FRN\_WINNT to the list.

### Client Development Tip



To get an idea of how to go about writing code for Content Manager, go through the sample code provided with Content Manager in C, which demonstrates various operations and is well documented. They are located in *ibm\cm\samples*.

An overview of the sample files to get you started:

## Sample source code and its description

**frnoexse.c** - Sample application code to locate items in the database that match specified attributes. The APIs used are SimLibLogon, SimLibFree, SimLibSearch, and SimLibLogoff.

**frnoexdo.c** - Sample application code using Folder Manager APIs to create a folder ItemID, create a document ItemID, store an object as part of the document, and add the document to the newly created folder. The APIs used are SimLibLogon, SimLibFree, SimLibCreateItem, SimLibCatalogObject, SimLibAddFolderItem, and SimLibLogoff.

**frnoexfo.c** - Sample application code using Folder Manager APIs to create a folder. The APIs used are SimLibLogon, SimLibFree, SimLibCreateItem, and SimLibLogoff.

**frnoexno.c** - Sample application code to add a note to a document. The APIs used are SimLibLogon, SimLibFree, SimLibGetItemAffiliatedTOC, SimLibOpenObject, SimLibSeekObject, SimLibCreateObject, SimLibWriteObject, SimLibCloseObject, and SimLibLogoff.

**frnoexre.c** - Sample application code to replace a previously stored object. The APIs used are SimLibLogon, SimLibFree, SimLibOpenObject, SimLibWriteObject, SimLibResizeObject, SimLibCloseObject, and SimLibLogoff.

**frnoexwf.c** - Sample application code using Folder Manager APIs. This program:

- Reads a mortgage number from stdin
- Establishes a session with the Folder Manager
- Gets a list of all existing index classes
- Gets information about the MORTGAGE index class
- Gets a list of all existing workflows
- Gets information about the MORTGAGEWF workflow
- Creates a document with index class SIM\_INDEX\_NOINDEX
- Stores an existing file as part 1 of the document
- Adds the document to the first workbasket in MORTGAGEWF
- Opens the attributes for the document
- Indexes the document into the MORTGAGE Index Class
- Assigns a value to the MORTGAGENUMBER attribute
- Closes the attributes for the document
- Gets a copy of the attributes to verify the changes
- Routes the document from workbasket 1 to workbasket 2
- Completes the workflow
- Gets the table of contents for workbasket 2
- Closes the table of contents for workbasket 2
- Logs off from the Folder Manager

The APIs used are SimLibLogon, SimLibFree, SimLibListClasses, SimLibGetClassInfo, SimLibGetAttrInfo, Ip2ListWorkFlows, Ip2GetWorkFlowInfo, SimLibCreateItem, SimLibCatalogObject, Ip2StartWorkFlow, SimLibOpenItemAttr, SimLibChangeIndexClass, SimLibWriteAttr, SimLibCloseAttr, SimLibGetItemSnapshot, Ip2RouteWiplItem, Ip2CompleteWorkFlow, SimLibGetTOC, Ip2CloseTOC, and SimLibLogoff.

### Further Reference:

For help regarding Client Development refer to the following manuals available with CM:

Application programmer's guide for Windows

Application programming reference, volume 1

Application programming reference, volume 2

URL : [http://www-4.ibm.com/software/data/cm/pubs/library\\_cm71.html](http://www-4.ibm.com/software/data/cm/pubs/library_cm71.html)

## Appendix A

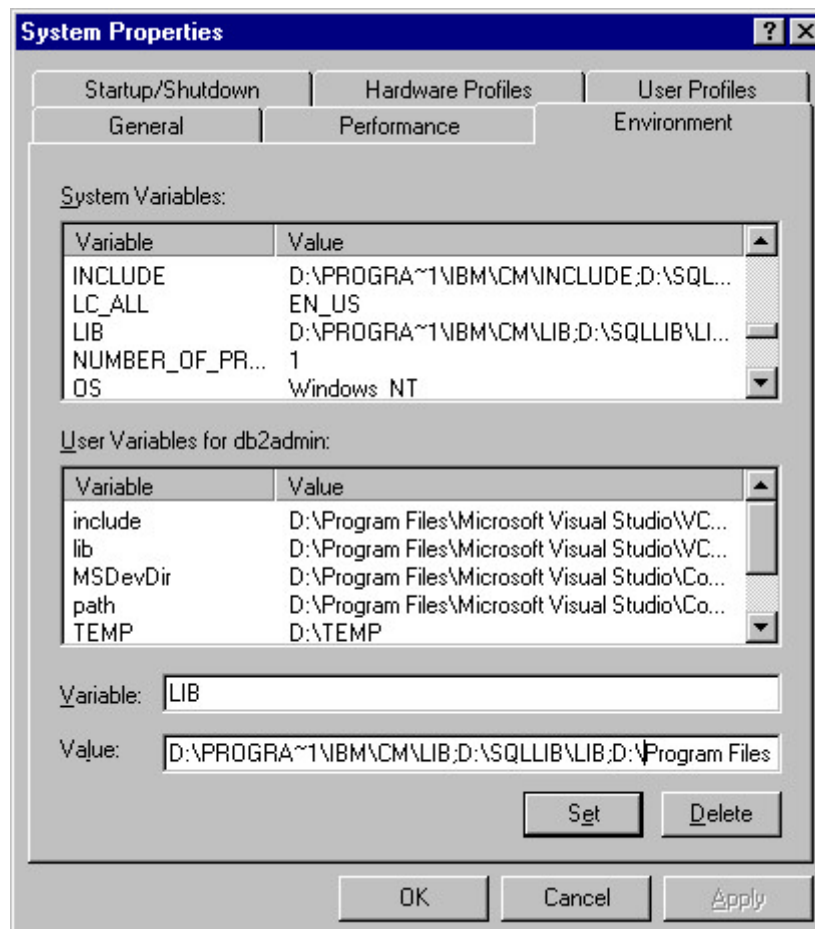
### Installing Content Manager

The following sequence needs to be followed while installing Content Manager

1. Install IBM DB2 Version 6.1
2. Install Visual C++
3. Install Content Manager 6.1

#### Important Points

1. There were problems running CM when a later version of DB2 was installed.
2. After installing DB2 , a new user is created db2admin which will act as the Content Manager Library Server Administrator ID.
3. After installing VC++, one needs to copy the INCLUDE, LIB and PATH variables to the system variables so that these values are available to all users. Go to **My Computer -> Properties -> Environment**

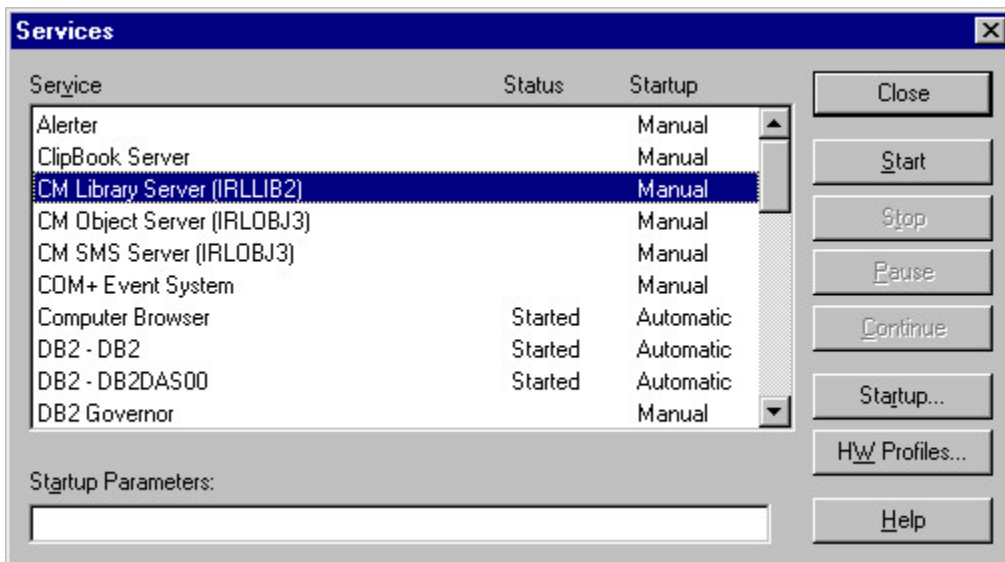


4. After creating the db2admin user in DB2 installation , 4 special privileges need to be given to the user.

See the **Content Manager Installation Guide** for further details.

## Starting the services

After Content Manager is installed, one needs to start the Content Manager Services to start the server. Go to Start->Settings->Control Panel -> Services and click on the appropriate service to start it.



Start the “CM Library Server” for starting the Library Server only.  
Start the “CM Object Server” and “CM SMS Server” to start the object server also.

## Network Table Setup

Any Content Manager Server (Library or Object) needs to know the location of other object servers and the Library Server. This table listing the locations for each server is known as the Network Table. This is stored in the file FRNOLINT.TBL in the CM root installation directory or in the Active Directory depending on the CM configuration.

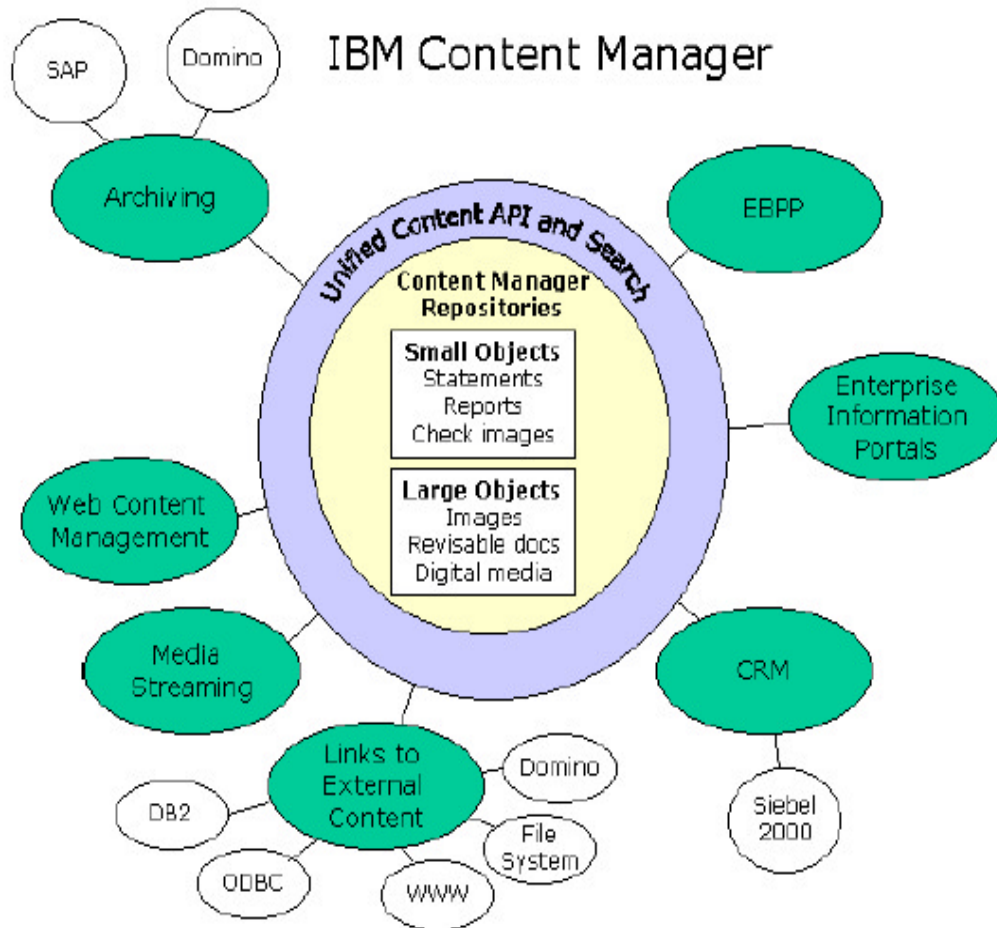
To generate the network table for a server, run the program **Utilities->Network Table Generation** and specify all the servers this server needs to talk to.

## System Administration Client

Run the System Administration Client to configure the Content Manager Library Server and Object Servers. All configurations for users, security, index classes, storage policies etc. can be done through the System Administration Client.

## Appendix B

### A Brief Discussion of features of CM: From the White Paper by Bruce Associates



**Figure 2. IBM Content Manager surrounds performance-optimized native repositories with unified search and access APIs for use with a wide variety of applications.**

## IBM Content Manager

Content Manager represents the unification of IBM's proven technologies for managing document images (VisualInfo/ImagePlus), digital multimedia (Digital Library), revisable content (Domino.Doc), host output (OnDemand), and federated searching (EIP), recognizing that e-business demands seamless



access to any type of content. The Content Manager strategy revolves around six basic principles:

**1. Provide scalable digital repositories optimized for any type of content.**

IBM Content Manager provides repositories performance-optimized for both large and small object types. One is designed for large objects such as images, fax, audio and video, revisable desktop documents and electronic forms, and the other for smaller objects such as host output, including statements, invoices, and reports, or check images. Both are enterprise-scalable and run a broad range of server platforms.

**2. Provide uniform access by business applications to information stored in the Content Manager native repositories – and external data sources as well.**

IBM Content Manager delivers information integration capability through a Unified Content API, in C++, Java, or ActiveX, enabling e-business applications with access to any type of content needed to support the business process. Search templates translate the index fields of existing content stores into a common vocabulary used in the query screens, and translates disparate file formats upon retrieval into web-viewable renditions. This gives IBM Content Manager a major advantage over other products that require existing applications to be rewritten in order to take advantage of cross repository searching, retrieval, and viewing.

The Unified Content API also dramatically simplifies integration of disparate external data sources in enterprise information portals, including portals for customer service, knowledge management, e-commerce, HR, or business intelligence. It allows users to perform queries across virtually any repository, including Content Manager, file systems, web sites, Domino databases, structured data in DB2, Oracle and other relational databases, and business intelligence metadata, consolidating results in a single hit list.

**3. Web-enable e-document search and retrieval.**

Content Manager offers a new out-of-the-box eClient that provides web-enabled search and retrieval from one or multiple Content Manager repositories. Using Java servlets on a web application server, the eClient provides a firewall-friendly “ultra-thin” client that runs in a web browser, and even allows users to email documents and launch applications for full function viewing as needed.

**4. Automate business processes that use e-documents.**

Workflow makes documents actionable, turning them from simply information into key components of e-business transactions. IBM Content Manager includes basic document processing workflow with native queuing and routing capabilities. For example, a workflow can be automatically initiated upon the arrival of a scanned object, fax or eform from the web for indexing into Content Manager. In addition, IBM Content Manager links with IBM MQSeries Workflow, which provides a solution-building framework that interconnects tasks and applications

throughout the enterprise, to achieve business process automation. Also, through built-in integration with native process automation in Siebel, SAP, Domino, and other major applications, IBM reduces the effort of connecting workflow and content management.

Following points are in relation to e-Business.

**5. Integrate with industry-leading Customer Relationship Management solutions**

Siebel Systems is the market leader in CRM solutions, particularly for large and midsize companies. IBM provides a tight integration of IBM Content Manager with Siebel 2000 eBusiness Applications. The IBM offering provides a unified logon to Siebel Call Center and IBM repositories, one-click access to repository documents from the Siebel desktop, integrated indexing of inbound correspondence from Content Manager queues and linking them with Siebel service requests, automatic creation and archiving of outbound correspondence, and integration of third party image capture subsystems. By document-enabling their familiar call center agent desktop, companies can maximize customer satisfaction and retention with less custom development and training.

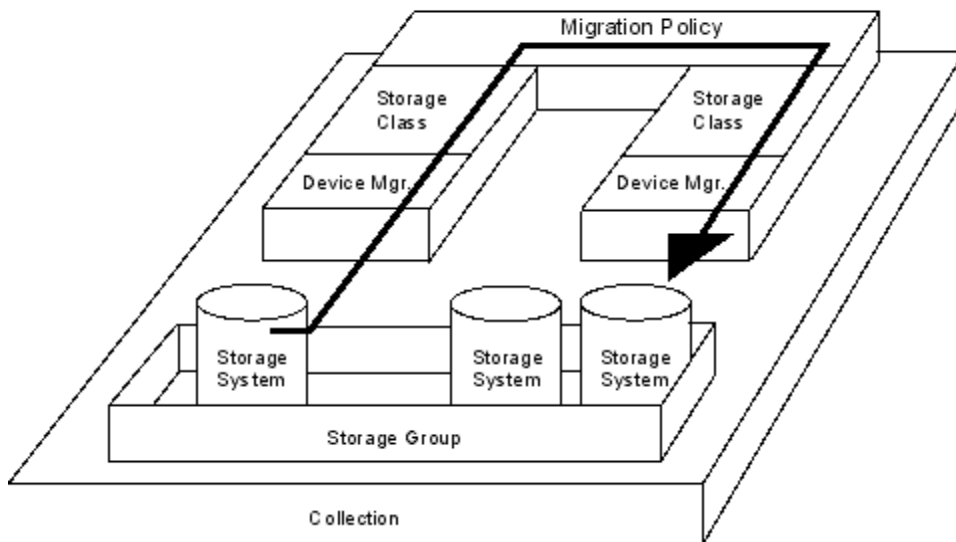
**6. Provide integrated archive solutions for leading business applications.**

IBM Content Manager provides data and document archiving solutions that integrate Content Manager repositories into the mySAP or Domino user environment. Both products significantly improve system performance and reliability of these critical user applications by offloading huge volumes of documents and historical data to the repository, leaving behind just a link for instant retrieval. The SAP version enables prioritized archiving of data as well as incoming or outgoing documents to disk, tape, or optical for performance, and integrates seamlessly via the standard Archivelink interface. The Domino version archives email and entire Notes databases, accessible from either Notes 5 or web clients.

## Appendix C

# Migration policies and Logging in Content Manager

## Section A: Migration policy



CM documentation states:

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.

The **migration policy** defines how long an object stays in a location and where the object will be moved to 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. You must associate the **storage class** with at least one **storage system** in the **storage group** for the **collection** to which the object belongs.

To migrate an object to another object server, specify a remote storage class as the final step in a migration policy.

Use the system administration program to create the migration policies that you need to fit the needs of your business. You can use the same migration policy for more than one collection.

In brief, the migration of objects in CM is only effected by time-based parameters, which shift the object after a certain time. No other criteria to control the migration of objects in CM is presently available.

## Section B: History Log

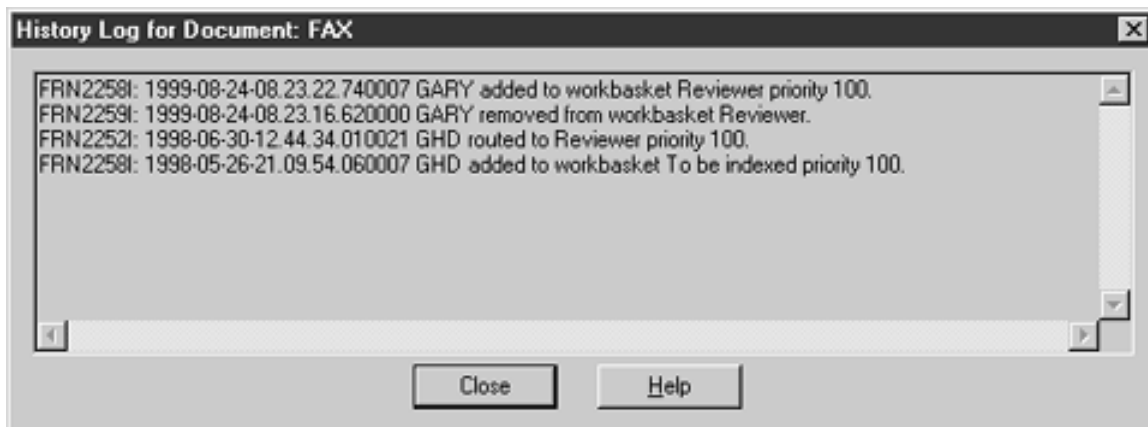
CM Manual says :

1. When items are processed through a workflow that has the save history log flag turned on, significant events that occur to each item are recorded in the events table, SBTEVENTS. The history log utility migrates these events to the history log on the object server. The history log for each completed item is migrated to the object server specified in the index class that is associated with the item.
2. Use the history log utility to migrate the history log from the library server to the object server for each completed item in a workflow.

### API Related

3. Use the **Ip2ListHistory()** function to display the log of activities recorded for the item that you specify. This log is a list of entries that document several actions that affect the status of the item.
4. Use the **Ip2WriteHistoryEvent()** function to add an event entry in the **history** log.

A sample view of the history log:



## Description of History Log

The History log stores the workflow history of the object. Whenever an object is moved from one workbasket to another or its priority in the workbasket is changed, an event is recorded in the history, which can be later viewed for examining the processing of that particular object. It does not store a log of all the operations done on items by a client; hence it may not serve the needs for Audit Trail or similar requirements.

## Appendix D

### Additional Products for Content Management from IBM

Components and complementary IBM Content Manager products include:

#### **IBM Content Manager: Organizing and digitizing enterprise information**

Whether video, audio, XML, image or PC files, multimedia assets can now be digitally transformed and archived and efficiently distributed over public or private networks. With IBM Content Manager, it is possible to deploy departmental or enterprise-wide solutions to support a wide range of business needs, from e-commerce catalogue enablement to high-volume document imaging for claims processing.

#### **Content Manager OnDemand: Fast access to large volumes of data**

The ability to capture, store and retrieve computer-generated information – invoices, customer statements, management reports and the like—is mission-critical to superior customer service. Content Manager OnDemand captures documents from their original application, then lets users throughout your company electronically access, distribute, print or fax the documents as needed. With OnDemand, paper and microfiche archives could become things of the past –minimizing search time and the storage required by reams of paper and stacks of microfiche. Additionally, OnDemand increases your e-business capabilities and encourages leading-edge customer service with options available for electronic bill presentment and payment solutions.

#### **IBM Enterprise Information Portal: Powering the critical content storehouse**

IBM Enterprise Information Portal (IBM EIP) allows users to search across and beyond multiple content repositories. Using IBM EIP, you can personalize your queries as you perform a federated search in a single request. It is then possible to extract pertinent information from documents, printed output or traditional databases. IBM EIP simplifies extensive searches across multiple servers and data sources for very specific information needs, from both IBM and non-IBM digital libraries.

#### **MQSeries Workflow: Optimizing processes and building profit**

IBM MQSeries® Workflow is a key complement to Content Manager. By acting as the centralized “backbone” for managing your business processes, MQSeries Workflow helps you design and control the overall flow of work across your organization to ensure that all tasks are prioritized and completed –on time, on target, and on budget.

**Domino.Doc: Collaborating virtually...and securely**

Domino.Doc™ is an out-of-the-box document management application that enables groups of individuals –authors, reviewers or managers –to collaborate on the same document or set of documents. Domino.Doc gives your organization the ability to capture, store, retrieve and distribute documents across virtual communities –from Lotus Notes® clients, Web browsers, or direct from ODMA enabled Windows® applications.

## Appendix E

### Useful References for Content Manager

Content Manager Home Page

<http://www-4.ibm.com/software/data/cm/>

Content Manager Library (Contains White Papers and brochures)

<http://www-4.ibm.com/software/data/cm/library.html>

Content Manger Technical Library (Contains all technical manuals)

[http://www-4.ibm.com/software/data/cm/pubs/library\\_cm71.html](http://www-4.ibm.com/software/data/cm/pubs/library_cm71.html)

### Important Guides for Content Manager

Application programmer's guide for Windows (Jan 01)

<http://www-4.ibm.com/software/data/cm/pubs/cm71/apgwin/frndmst.htm>

Planning and Installing Content Manager

<http://www-4.ibm.com/software/data/cm/pubs/cm71/cminstal/frnp9mst.htm>

System Administration Guide

<http://www-4.ibm.com/software/data/cm/pubs/cm71/sysadm/frns2mst.htm>

### Related Links

More than 200 white papers on Content Management at Itworld.com

<http://www.itpapers.com/cgi/SubcatIT.pl?scid=178>

Companies need new ways to manage their data: ItWorld

<http://www.itworld.com/AppDev/146/IWD010416hndataman/>

Microsoft's answer to Content Management

<http://www.microsoft.com/technet/ecommerce/contmgt.asp>

Content-Wire

<http://www.content-wire.com/Home/Index.cfm>