

The Alchemy[®] Database

An Open and Extensible Architecture for Efficient Management of Archived Data

Abstract:

This paper discusses key concepts regarding the Alchemy database.

- The corporate challenge with archived data management
- The database architecture and data structures
- Compatibility with industry standards
- Database storage options
- Benefits of the database



Information Management Research

6025 South Quebec Street, Suite 260
Englewood, CO 80111
303-689-0022

www.imrgold.com

Summer 2003 update for Release 7

TABLE OF CONTENTS

The Alchemy Database

An Open and Extensible Architecture for Efficient Management of Archived Data

Section 1

- Introduction p. 3
- 1.1. The Corporate Challenge with Archived Data Management
- 1.2. The Solution: Alchemy

Section 2

- Database Architecture p. 4
- 2.1. Client/Server Implementation
- 2.2. Reliability
- 2.3. Multiple databases
- 2.4. Simultaneous database users

Section 3

- Data Structures p. 6
- 3.1. B-tree Data Structures
- 3.2. Database File Structures
- 3.3. Database File Storage
 - 3.3.1. Example where documents are not copied to the data container
- 3.4. Database updates
- 3.5. When additional .DAT and .PRO files are created
- 3.6. Database backup and recovery
- 3.7. The compound document structure
- 3.8. Database performance considerations
 - 3.8.1. Query (Read) operations
 - 3.8.2. Write operations

Section 4

- The Database Query Engine p. 10

Section 5

- Compatibility with Industry Standards p. 11
- 5.1. ODBC SQL interface to Alchemy
- 5.2. COM interfaces to data structures
- 5.3. WIN32 file system
- 5.4. Option to store documents outside the container file

Section 6

- Database Storage Flexibility p. 12
- 6.1. File Server Hard Disc
- 6.2. RAID (Redundant Array of Inexpensive Discs)
- 6.3. Workstation Hard Discs
- 6.4. CD Writer/Reader
- 6.5. CD-ROM Tower
- 6.6. CD-ROM/DVD Jukebox
- 6.7. Optical Libraries
- 6.8. Combination of Devices

Section 7

- Benefits of the Database Summarized p. 14

1. Introduction

This paper covers the following key concepts regarding the Alchemy database:

- The common problem with unstructured data management
- The database architecture and data structures
- Compatibility with industry standards
- The benefits of the database summarized

1.1. The Corporate Challenge with Archived Data Management

Before explaining the Alchemy database, the paper will discuss the problems Alchemy was designed to solve.

Most of the knowledge and information created by companies resides outside of structured databases or outside the applications that created the data, and are labeled “unstructured data”. Getting control of this unstructured data is widely recognized as the key to accelerating necessary enterprise operations.

The unstructured data at most organizations is typically made up of many different types of files and formats: paper, microfilm, scanned images (TIFFs), computer report files (COLD), Adobe Acrobat PDF files, text files, Microsoft Office documents, color images, video, audio and other binary files.

The Internet explosion has also created a new class of unstructured data types that are threatening to overwhelm information management systems:

- Email, especially with attachments
- HTML files

Relational databases such as Oracle and Microsoft SQL have introduced methods to integrate and manage unstructured data. While it is possible to integrate unstructured data content with a relational database, these methods are often too complex for many companies to implement, and implementation can require significant computing and programming resources.

1.2. The Solution: Alchemy

Alchemy was introduced in 1994 as a Microsoft Windows 3.1 software product that enabled companies to efficiently manage, archive and distribute unstructured data, using the new CD recordable technology as the new paradigm for storage. In subsequent product releases, IMR expanded Alchemy to become today's 32-bit database manager of all unstructured data types utilizing any storage platform compatible with Windows NT, Windows 2000, and Windows XP Professional, and to manage ever-larger databases that used to reside only on mainframe systems. There are now over 10,000 installations of Alchemy in over 40 countries, at companies and agencies of all sizes, including many Global 2000 companies.

With Alchemy, managers can:

- Effectively capture and manage all unstructured data in one location.
- Create a single point of access to unstructured data, regardless of the format of the original information.
- Deploy solutions in days, not weeks or months. IMR provides highly developed user interfaces that can be deployed immediately to access the database.
- Safeguard information assets by archiving them into Alchemy databases.
- Utilize cost-effective storage media such as CD or DVD.

The most popular unstructured data types under Alchemy management today are:

- ASCII files that originate as reports or print files from other databases (sometimes referred to as computer output, COLD or ERM, enterprise report management)
- Microsoft Office documents (Word, Excel, PowerPoint)
- Scanned images with or without OCR text
- Adobe Acrobat Portable Document Format (PDF)

- AutoCAD files
- And hundreds of other file types

Alchemy databases are also supported by leading document capture companies such as Adobe, Cardiff Software, Captiva, Kofax, Konica, Input Software, Readsoft and ScanSoft. Documents and data produced by their applications are integrated with the Alchemy database.

IMR also provides a Web Server that will support Alchemy users through an Internet Explorer-enabled web client. This gives Internet or Intranet users access to all of the features and functions of Alchemy through an easy to install and maintain client implementation.

2. Database Architecture

IMR developed its own object database architecture with hierarchical data structures and is now on the 7th generation where it is proven at thousands of installations every day. The large installed base makes the Alchemy architecture one of the most widely used databases for unstructured data storage/retrieval applications.

In the unstructured data and document management industry, IMR is unique. Other software companies in this industry build their products on top of Microsoft SQL Server, Access, FoxPro, Oracle, Sybase, Pervasive (formerly Btrieve) or other databases. While this may have helped them somewhat in terms of scalability, this strategy had its downsides: the increase in system complexity, the higher total cost of ownership and the lengthened deployment time.

IMR believes this reliance on relational databases for unstructured data management is responsible for the lukewarm acceptance or outright failure of many enterprise wide document management initiatives. In comparison, companies who adopt the Alchemy database architecture find it to be simple to install and quick to deploy, and completely optimized for performance with unstructured data types. Furthermore, the basic architecture is open, extensible and scalable, and will continue to be a superior means for unstructured data management.

The performance advantage of the Alchemy database over a relational database is easily understood. In Alchemy, a query of all indexes and subsequent retrieval of documents for viewing can usually be resolved in one, or maybe two passes of the index at most. In a relational database, the same operation will take several passes assuming the query is optimized. What does this mean to the person who uses Alchemy daily? The knowledge worker finds and uses information faster; the IT department can offer more performance without adding expensive mega-servers with mega-storage; and everyone avoids the complexity and maintenance issues associated with a relational database.

2.1. Client/Server Implementation

Internet/Intranet: The Alchemy Web Server implementation is true client/server. Client requests are handled at the server and directed to the appropriate database files.

LAN/WAN: An Alchemy client on a Windows LAN/WAN opens a direct connection to the database files for read and write operations. The current architecture is optimized to handle hundreds of simultaneous requests.

2.2. Reliability

The Alchemy database offers a high degree of reliability and data integrity. Transactions such as data field entry and adding a file or folder are written immediately to disc. Temporary interruptions such as power dropouts will not cause a data loss since Alchemy maintains backup files to restore data.

Data field transactions are committed to the database in real-time. If the system crashes or the power goes out, the database can be restored up to the moment of the power failure. For example, when an end user completes data entry into a profile field, the data is immediately written to disc and the user

moves to the next field or record. When the user creates a new folder, that information is also written immediately to disc.

Documents are added to the database in two steps; it is important to understand this process in order to appreciate the database's ability to recover.

1. First, when documents are added, pointers to each document are written to disc. The documents being added to Alchemy are coming from somewhere on the user's system or network, and should not be purged by the user until a successful Build operation is completed.
2. The Build Database operation copies the document into the .DAT file and creates the searchable index. During the Build operation, the .ALB Database Backup File is created. If a database build process is interrupted or cancelled, Alchemy uses this file to restore the database to its prior state. A database log records error messages and any missing or skipped files for each Build operation.

Reliability is also improved through support for multiple databases. A failure in a single database can have little or no impact on the larger system.

2.3. Multiple databases

Alchemy database technology enables administrators to distribute data storage across multiple physical databases. An Alchemy installation can grow in a virtual sense, while limiting risk and maintaining the benefits of centralized control. The customer's server capacity is limited only by the amount of hardware storage available. Scaling the database storage is invisible to the end user.

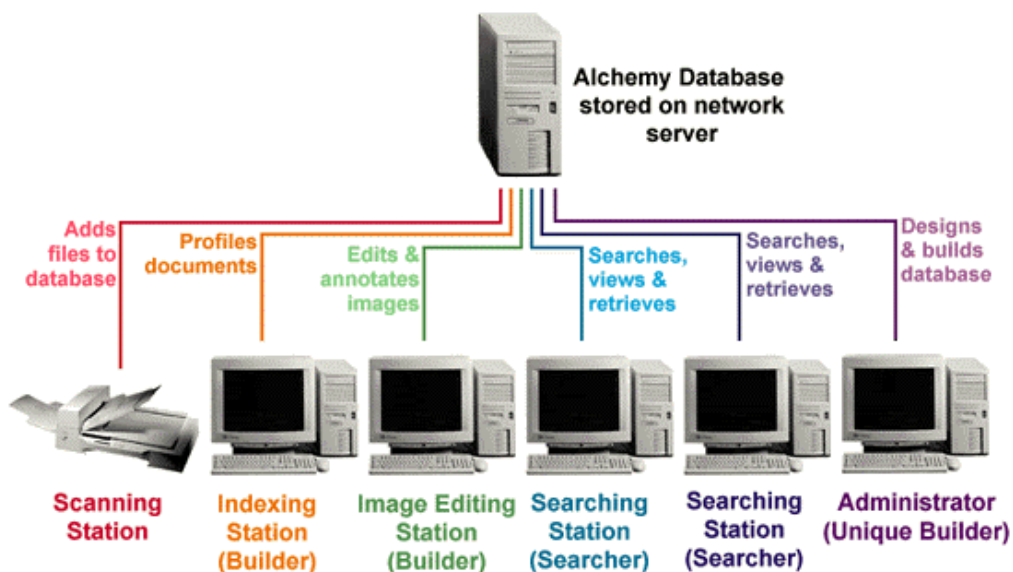
Multiple databases offer administrators a wide range of design flexibility. One server can host different sets of databases. Databases can be targeted towards specific user groups, such as Accounts Payable, Accounts Receivable, Sales, Marketing, Support or Engineering. This flexibility enables the administrator to centrally control database configuration parameters such as the content indexing methods, security groups and data views.

Another advantage of multiple databases is to keep each individual database smaller, which can improve backup and recovery time, and minimize the impact of a disaster. If there is a failure of one database, the other databases are not affected and those users will not experience downtime.

2.4. Simultaneous database users

The LAN/WAN architecture can support hundreds of simultaneous writers and readers into a database under the right conditions. The Internet/Intranet architecture is currently read-only and can support hundreds of simultaneous users under the right conditions.

Here is a typical network installation of Alchemy with a mix of database access licenses.



3. Data Structures

3.1. B-tree Data Structures

Alchemy uses an internal B-tree data structure method to store data. This method is widely used in the database industry. B-trees are special data structures for quickly manipulating data files. B-tree algorithms were developed by various computer manufacturers and independent research groups beginning in the late 1960s, and are now the basis for many commercial database systems.

A B-tree (which is computer science jargon for a "balanced tree") is similar to a SELECT list, but with some additional "pointer" information to allow very fast lookups, insertions, and deletions anywhere in the list. A B-tree is essentially a sorted list of all the item identifiers from a data file.

3.2. Database File Structures

An Alchemy database consists of several important files that are created through the Alchemy GUI or the APIs.

A. Data container (.DAT)

The data container contains a copy of each document that was added to the database. The document is stored and retrieved in its native binary format. There is an option for document compression. ASCII files are compressed up to 10:1. The .DAT file can be password-protected and encrypted with 128-bit encryption.

B. Full-text index (.NDX)

Full-text indexing is an integral component of the database. IMR developed its own full-text engine to insure superior compression and query optimization. When a text file is added to the database, Alchemy uses file filters to open and extract the contents of the file and to add that text to the .NDX file, a highly compressed file containing every word that was extracted from every document. This file is then utilized for full-text retrieval, a powerful tool for data mining and knowledge sharing. There is one .NDX file per database. This file cannot "span" disc drives.

The index is created automatically and can be modified by the user to exclude words or characters, to index pure numbers and other standard indexing design criteria. **IMR developed unique text index compression algorithms, resulting in very low file size overhead typically 3-5% of the original text file size.** This compression is directly related to some of Alchemy's advantages in retrieval performance, storage management and CD/DVD portability.

C. Profiles (.PRO, .TPR)

An important characteristic of the Alchemy database is the ability to store data properties for each item in the database. The .PRO container is a file that contains the profile fields for every document. In Alchemy, profile fields are used to store data as a property of the document and as a method to query the database. This terminology has its origin in document management. Some profile fields are created automatically for every document (filename, file format, file date, etc). Each document has a unique ID number field. Custom profile fields can be added by the user, and automatically populated from other data sources through the DataGrabber interface or APIs, or manually populated through the drag/drop GUI.

Profile fields can be indexed and queried using Boolean searches. The database supports multiple field queries using AND/OR separators, and also supports a single query across profile fields and the full-text index. **Profile fields in Alchemy have important advantages over SQL data fields.** Each profile field can store up to 4,096 bytes. An Alchemy database can have over 100 profile fields.

The .TPR file stores profile data before it is indexed by the database during the Build database operation.

D. Hierarchical file structure (.TID, .TOB)

These files contain the hierarchical file and folder structure, and are created when files and folders are added to through the Alchemy Contents GUI or through the API Add Files function. Alchemy supports the nesting of folders.

E. Database definition file (.ALD)

This file contains database definitions. To access an Alchemy database, the system must point to this file.

F. Database resource files (.RID, .ROB)

These files contain identifiers for profile field templates, field definitions, and lookup tables.

G. Annotations (.ANT)

If the annotations tool in the Alchemy GUI is used to create annotations for a document, they are stored in the database as a document property.

H. User Settings file (.INI):

The Alchemy .INI file contains user-definable settings for the database, such as preferences, last database opened, fields to display in the list view and much more.

I. Database Log File (.LOG):

The log file contains a history of all database activity and errors. A separate log file is created for DataGrabber operations.

J. Database Backup File (.ALB)

During the Build operation, the .ALB Database Backup File is created. If a database build process is interrupted or cancelled, Alchemy uses this file to restore the database to its prior state.

3.3. Database File Storage

An Alchemy database is self-contained and portable. An Alchemy database does not require a dedicated server. Therefore the database files can be stored on any Windows-compatible disc drive and opened by anyone who can access that drive. Databases can exist on a wide variety of storage devices, from laptop CD-ROM drives or desktop hard discs, to huge optical libraries attached to an Windows NT, 2000 or 2003 server, DVDs or RAID.

To best understand how database files are stored, let's look at a typical database called "Reseller." This database contains several document types: Word, Excel, PowerPoint, AutoCAD, Acrobat, TIFF, COLD, color images and more. The database options used were:

- Store the database on an NT, Windows 2000 or Windows 2003 server shared hard disc.
- Copy the documents into the data container.
- Compress the data container.
- Store seven profile fields per document.
- Index all text documents
- Data size before compression: 68,876,663 bytes
- Data size after compression: 48,971,776 bytes

Here is the database root:

Name	Size	Type
Reseller		File Folder
Reseller.ald	1KB	Alchemy Database

Inside the database folder:

Name	Size	Type
Data000		File Folder
reseller.alb	1KB	
reseller.log	1KB	Text Document
reseller.lok	1KB	LOK File
reseller.ndx	272KB	
Reseller.rid	1KB	
Reseller.rob	16KB	
Reseller.tid	7KB	
Reseller.tob	88KB	TOB File
reseller.tpr	16KB	

Inside the Data000 folder:

Name	Size	Type
00001.dat	47,824KB	DAT File
00001.pro	16KB	

This file structure is maintained wherever the database is stored, with two exceptions. 1) When a database is archived to a removable disc, there is an option that will move only the .DAT and .PRO files to the disc, while the other files remain on the hard disc. This is a popular option for database files that are updated or accessed frequently. 2) When the documents are stored outside the data container, the .DAT file is essentially empty.

3.3.1. Example where documents are not copied to the data container

This database was created without copying the documents to the data container. Note the difference of file size in the DATA folder. That is because there are no documents inside. The database files are pointing back to the documents on the original disc drives.

Name	Size	Type
00001.dat	1KB	DAT File
00001.pro	16KB	

3.4. Database updates

When new data is added to the database, the data is merged into the appropriate database file. For example, if one were to add a new document and enter new data into its profile field, the document is compressed and added to the .DAT file; the profile field is added to the .PRO file. If full-text was extracted from the document, the new text is merged into the .NDX file.

3.5. When additional .DAT and .PRO files are created

When the database is built, Alchemy uses a temporary buffer on the hard disc to compress and index the documents before they are copied to the .DAT file. If the buffer size is exceeded, Alchemy will break the .DAT and .PRO files, and start new ones with sequential numbers. Buffer size is user-configurable. This architecture is what enables databases to be spanned across multiple archive discs, and is also integral to Alchemy's intelligent CD or DVD-writing capabilities.

3.6. Database backup and recovery

Given the straightforward file structure and file storage methods of the database, it follows that an efficient database backup and recovery strategy can be easily implemented. It is only necessary to backup the files stored on the hard disc that have changed. Database files archived to a

secondary storage device such as CD-R, DVD-R or tape are already “safe and secure”, needing no further backup.

Alchemy also maintains several database files on the disc where the database is located that are constantly updated with new data and document information. These files can be used to recover a database in case of a power outage or disc disaster, in most cases up to the point of interruption.

Of course, the user should always employ a good server backup strategy. The best strategy to guard against data loss caused by disc failure is to use RAID to replicate these key database files on one or more discs.

3.7. The compound document structure

The database can store documents together in a relationship called compound documents.



In this example, a bank statement is the parent document and the check images are the children. A query that returns any one of the documents will display this relationship. A common use is to relate a text document with images. Each document is still stored separately in the database and can be accessed apart from the compound structure.

3.8. Database performance considerations

It is important to understand where the advantages or the potential bottlenecks are in an unstructured data management system like Alchemy.

3.8.1. Query (Read) operations

In the Alchemy architecture, query operations such as full-text query or profile field query are resolved extremely quickly, as they are performed on static index files that do not require a join or merge operation. The database is capable of supporting hundreds of query transactions per second under the right conditions. The number one performance bottleneck for retrieval is the size of the documents that are to be viewed by the end user. When you think of retrieving data from a relational database, usually the data is quite small in comparison to the large binary files such as multi-page TIFFs that are managed by the Alchemy database. Accordingly large files require more bandwidth to transfer to the client for viewing and retrieval.

The system hosting an Alchemy database should be designed with sufficient bandwidth and storage speed to meet the needs of the end user, based on an analysis of the document types and the predicted pattern of access. The database itself has been optimized to reduce the amount of data transferred to the client. However there is still a predictable amount of overhead each time a document is opened by a client. This overhead is document-specific and can be analyzed for you by the IMR Professional Services group or an experienced Alchemy reseller.

3.8.2. Write operations

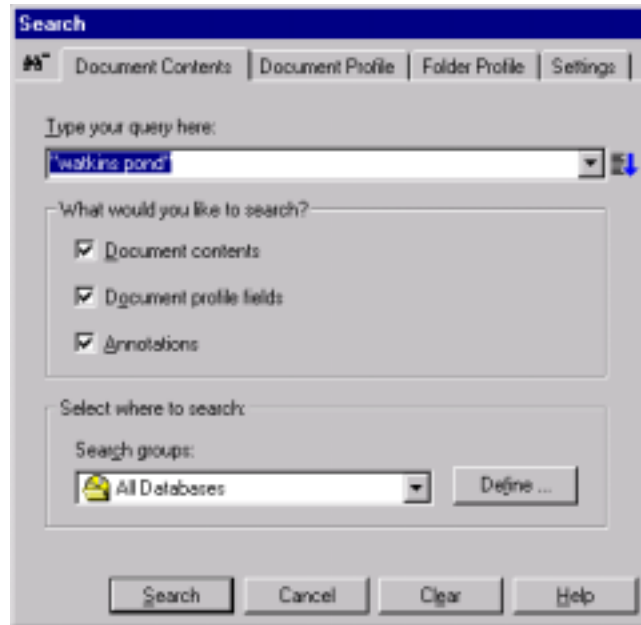
Clients writing to the same database at the same time compete for the .LOK file. The database handles all write requests in the order they are received, and displays a database busy message until the .LOK file is freed. Ad hoc write operations are very fast: examples include dragging and dropping some files into the database, updating a profile field, adding an annotation, adding a folder, scanning into the database with the IMR Scan interface, or deleting a file/folder. The database is capable of supporting hundreds of these transactions per second under the right conditions.

Production-level write operations consume more time: examples include building indexes, DataGrabber build operations that import large ASCII files, and batch file operations. It is

recommended that these operations be performed offline, or at a time when other users with write access are not logged into the database.

4. The Database Query Engine

IMR developed a very powerful query interface to the database. This interface is accessible through the Alchemy Search client or programmatically through the APIs.



Here is a list of database query capabilities.

1. Full text query – document contents and OCR text
2. Document profile field query – single or multiple field queries
3. Folder profile field query – single or multiple field queries
4. Annotations text query
5. Word or phrase query
6. Single query across full text, profile fields, folders and annotations, or any combination thereof
7. Query across a group of databases
8. Boolean AND/OR
9. Wild cards
10. Fuzzy word search
11. Relevancy ranking of search results
12. Saved queries
13. Drill down query of search results
14. Date range search > = <
15. Numeric search > = <
16. Proximity search/adjacency searching
17. The search interface also displays a word wheel that exposes the data for every indexed .PRO field and the full text .NDX file.
18. Query results can be published as HTML pages through the Alchemy Web Server.
19. Searches across multiple CDs or DVDs.

5. Compatibility with Industry Standards

While the database has many unique features and is virtually maintenance-free, it is also open to other applications through two industry standards: COM and the Windows file system.

5.1. COM interfaces to data structures

The database also has COM interfaces that are compliant with the widely adopted Microsoft Component Object Model. IMR has exposed the significant data structures such as the data containers, the profile fields and the full text index. Any COM-compliant programming environment can communicate directly with the database files and bypass the proprietary Alchemy GUIs.

5.2. WIN32 file system

The Alchemy interfaces support document and profile field retrieval from the database to a directory on a Windows drive. It is important to note that Alchemy stores all documents in the native file format, so that a document retrieved to the Windows file system retains all original attributes and can be opened in the native application.

Profile data can be exported as an ASCII delimited file or a formatted ASCII report file. Documents can be exported as single files or concatenated into one file.

5.3. Option to store documents outside the container file

There is an option to create an Alchemy database without copying the documents to the .DAT file. This was developed for two purposes: 1) to expose the documents directly to the operating system, and 2) to use Alchemy for indexing and retrieval but leave the documents on a legacy storage system such as magneto-optical.

6. Database Storage Flexibility

A unique capability of the Alchemy database is its portability. This capability enables storage of Alchemy databases on any media type compatible with the NT file system. **No other database gives the customer so many storage options.** The benefits of this are:

- Flexibility in choosing the best storage option for each database.
- Ability to optimize the conflicting needs of speed, size, and cost.
- Ability to relocate the entire database to another drive.

This section will examine some of the most popular storage methods for Alchemy databases, and provide recommendations as to when they should be used.

6.1. File Server Hard Disc

The most straightforward method of storing an Alchemy database for shared use is to store it on the hard disc in the NT Server, Windows 2000 or Windows 2003.

Advantages:

Using the server hard disc is the most direct method for accessing an Alchemy database on a network, and hard disc is the fastest storage medium available, making this solution the fastest available.

Considerations:

The disc capacity on the server disc is often limited, which could be a problem if the Alchemy database(s) is large. Also, some server disc capacity must be left for Alchemy to use as temporary space to add files to a database.

6.2. RAID (Redundant Array of Inexpensive Discs)

To oversimplify, a RAID system is just a big hard disc system with greater reliability and more capacity than a standard hard disc. In general a RAID system will provide very fast access times and transfer rates, and will allow for the storage of larger database(s).

Advantages:

RAID systems have fast access times, and higher reliability and capacity than a standard hard disc. RAID is an excellent choice for high-traffic applications.

Considerations:

They are more costly than standard hard discs.

6.3. Workstation Hard Discs

Alchemy databases can reside on any local hard disc. This could be a Windows NT, 2000, 2003 or XP Professional workstation.

Advantages:

A local hard disc is the fastest device type available, so it is always a good choice for storing an Alchemy database.

Considerations:

If the Alchemy databases are large, disc capacity can be a problem. Also, if the database needs to be shared with others on a network, it may become unavailable or slow to access.

6.4. CD and DVD Writer/Reader

The database can be stored directly to a writable CD/DVD platter in an inexpensive CD/DVD writer. The CD/DVD can be read from that drive, or removed and read in other CD-ROM or DVD-ROM drives.

Advantages:

Low cost and universally compatible method for archival or distribution of databases. Permanent, non-erasable archive of databases.

Considerations:

650 MB capacity limitation on a CD. Cannot access data on CDs or DVDs that are not mounted. Local CD-ROM and DVD-ROM drives can be shared on the network but the performance is slower than hard disc.

6.5. CD/DVD-ROM Tower

A CD/DVD-ROM Tower is a network-attached device where each disc is stored online in a reader. Dozens of readers can be stacked in some CD/DVD towers.

Advantages:

Permanent, non-erasable archive of databases. Higher capacity for larger Alchemy database(s), which can be spanned across media. High on-line capacity and faster access times than jukeboxes because of multiple readers. Access speed can rival hard discs. Portability – a CD/DVD can be removed from the tower, distributed to other locations and read elsewhere.

Considerations:

The performance of a CD/DVD-ROM tower in the real world will depend on many factors. Where the tower is located is one factor. An example of a good environment for a CD/DVD-ROM tower would be a customer service application where recent customer invoice data is stored on a few CDs or DVDs and accessed frequently, and those discs can be kept constantly mounted in readers.

6.6. CD-ROM/DVD Jukebox

A CD-ROM or DVD Jukebox is a network-attached device that stores a very large number of CD-ROM/DVD discs, with a small number of CD-ROM or DVD readers. Systems are available with 100-1,000 CD/DVD slots and one to several readers. The advantages of jukeboxes is the ability to hold vast databases at the lowest cost per gigabyte available. Additional device drivers may be required.

Advantages:

Permanent, non-erasable archive of databases. Huge capacity for very large Alchemy database(s). Lowest cost per gigabyte. Portability – a disc can be removed and read elsewhere.

Considerations:

The slowest medium for storing Alchemy database(s). The small number of readers can mean constant disc changes, which can result in long waits for data as the number of users grows. Should be limited to applications with very low frequency of access. The performance can be improved by adding more readers and by utilizing hard disc cache.

6.7. Optical Libraries

Alchemy databases can be written to and accessed from NT or Windows 2000/2003-based optical libraries that are mounted on the network as a drive letter. Additional device drivers may be required.

Advantages:

Huge capacity for very large Alchemy database(s). Much faster access than CD-ROM or DVD jukeboxes.

Considerations:

Cost is considerably higher than CD/DVD based systems. Magneto-optical discs are erasable, which may not be suitable for archival applications.

6.8. Combination of Devices

The Alchemy database index files can be stored on the server, and the data containers stored on a network-attached jukebox or tower. This allows for extremely fast search results of huge text databases or COLD data albeit with slower document viewing. This allows high-speed updates to the database.

7. Benefits of the Database Summarized

√ Increased Performance

When it comes to the search and retrieval of unstructured data, the Alchemy architecture is inherently more efficient than the combination of a relational database and a file system. The database structures are optimized for full-text operations and the seamless integration of document properties and metadata. Originally developed for slow access CD-ROMs, the query and retrieval performance is even more impressive when the databases are stored on RAID or other hard discs.

- **Searches across large databases can be completed in one or two passes.** Compare this to several passes typically required in a relational database.
- A query is done on static, flat-file index files, there are **no table join or merge operations** required.
- **A single query can automatically search across** the full-text indexes, profile fields and annotations in the same operation.
- A query across multiple, large databases can be **completed in a few seconds.**
- **Full text indexing is completely integrated** into the data structures, not implemented as a separate module like other products.
- **Superior compression of text indexes.** Alchemy's index size is typically compressed to 3-5% of the original text documents, compared to 25-100% for other text indexing products. This has two particular advantages: 1) a significant reduction in the storage space required for the database; and 2) inherently more efficient query operations.
- **File restoration to a local drive is not required to view the file contents.** The database opens a stream to the document inside the container file. This is especially important for multi-page files. Compare this to other databases, which may require that the entire document file be restored from the storage drive to the local hard drive before the contents can be viewed or otherwise manipulated by the client.

√ Powerful Document Security

- The database has **several security layers available.**
 - Password protection of the database.
 - Security groups within the database, with administrator rights and individual/group rights granted at the database, folder or document level.
 - Enforced annotations to secure sensitive, redacted documents.
 - Integration with Microsoft's Active Directory resource management system
 - Option to apply 128-bit blowfish encryption to the database files.
 - Option to disable document printing, copying or retrieval from the database.

Starting with Alchemy Premium version 7.3 a new security model may be optionally used with Microsoft networks. This new mechanism is modeled after the Microsoft Windows Security Model. This model leverages the security architecture that the LAN Administrator has designed and implemented so that the same access control mechanisms can be applied to the Alchemy environment. It allows authenticated Window principals discretionary access to Alchemy databases.

Integration of Microsoft Windows Security Model with Alchemy is dynamic and can be changed as well as turned off by the Alchemy Database Administrator on a Database level. It

is only available for databases managed by the Alchemy Premium version 7.3 or greater and it does not require changes of existing Alchemy security groups or Microsoft Windows security groups and/or users.

The Alchemy Premium security integration layer defines dedicated security objects call 'roles.' The names and semantics of the roles are driven by the business context needed for each installation.

- Documents and data stored in the compressed database files can be **accessed only through the secure database interfaces**. They cannot be:
 - Opened for viewing or printing.
 - Copied or retrieved.
 - Deleted by unauthorized users.
 - Modified in any way by unauthorized users.
- This method is **much easier to secure and manage**, when compared to other systems that store the documents out on a file server. In those systems, if the file server security is breached, the intruder has full access to the documents.

√ Lower Total Cost of Ownership for IT

- **Maintenance-free databases.** This is not a relational database and therefore the database does not require the same level of maintenance, repair, optimization and synchronization.
- **Lower disc space requirements for data backup.** Alchemy database files are compressed.
- **Reduced training costs for end users.** IMR provides highly developed end user interfaces for the database through the Alchemy GUI or ActiveX controls.
- **Very fast deployment of solutions.** Typical users are deploying their first Alchemy database within a few hours of installation.
- **Reduction in database administration costs.** The database management is GUI-driven and does not require relational database management or programming skills.
- **Simple, unified upgrade strategy.** The database and the user interface versions are always in sync; there is no need to worry about compatibility with a separate database.
- **Utilization of inexpensive secondary storage options** such as CD or DVD.



About IMR

IMR is the premier developer of document management software that gives every organization the power to retain and access all of their vital documents. This can include scanned documents, electronic documents, forms, enterprise reports, CAD/engineering drawings and email — the information generated by everyday business processes. IMR software is used by thousands of organizations to easily store, access and utilize these documents. Founded in 1992, IMR is a privately held corporation based in Englewood, Colorado and serves customers in 42 countries. IMR made its debut in 2001 for Software Magazine's "Software 500" list, the world's most significant providers of software and software-related services.

© Copyright 2002 Information Management Research, Inc. All rights reserved.