IRON MOUNTAIN®

GUIDEBOOK TO LONG-TERM RETENTION

PART 4: MANAGEMENT & DISPOSITION IN DATABASES & STRUCTURED APPLICATIONS

WHITE PAPER

CONTENTS

Structured data is generated by transaction-oriented enterprise applications such as enterprise resource planning (ERP), accounting, manufacturing or human resource systems. By definition, structured data is comprised of discrete pieces of related information stored in a well-defined format, such as the rows and columns of a relational database. Official business records are often comprised of multiple structured data points. As such, organizations must retain structured data transactions for legal, regulatory or business reasons, in addition to its use in analytics for competitive advantage, increased customer experience and more.

Managing long-term records in structured application presents several unique challenges for RIM and IT professionals. This white paper discusses a number of key issues and outlines several strategies for archiving and disposition of long-term records in databases and structured applications.

CHALLENGES OF MANAGING RECORDS IN STRUCTURED SYSTEMS

CHALLENGE: DEFINING AND SEGREGATING RECORDS IN A DATABASE

The concept of a record differs when comparing unstructured and structured content. When looking at unstructured data, such as a .pdf image of a scanned invoice, the data elements are contained in the record itself. In a structured database, the record is comprised of multiple data elements that may be stored in separate tables within one or more repositories, along with the database relationships linking the elements together. To create the record, an automated process executes queries that assemble the appropriate related elements.

Often reports or outputs from queries provide evidence of decisions or business processes. The outputs with business or legal needs should be managed as records per your organization's records and information management policy. The challenge comes with managing all the discrete data elements that are insufficiently complete to be regarded as records. In addition, often "summary" reports generated from applications and queries in the normal course of business are considered transitory and not subject to retention as a record.

A common way to distinguish potential records in a database application is by marking reports as "read-only." For example, when a payment has been received, the invoice status changes to "closed," indicating no further changes can be made. Separating records from non-records is achieved through the use of the underlying application or an appropriate SQL query that identifies

these types of read-only reports. Policies for retention and disposition can be applied and the data elements taken together can be considered as a record candidate.

CHALLENGE: STRUCTURED DATA CAN BE "IN MOTION"

In the course of routine business and IT operations, data may be moved and copied from one storage medium or repository to another. This process may occur because of:

- Limited storage capability and performance degradation. As databases expand, storage costs go up and end-user response times lengthen. IT may move rarely-used data off of production systems into an archival solution.
- Platform consolidation and obsolescence. Databases and structured applications may be consolidated onto a smaller number of servers and repositories to improve operational efficiency, reduce software license expense, lower administrative costs or to avoid technology obsolescence. Software and hardware obsolescence can be an impediment to accessing records if the legacy application is no longer available. (See the white paper: "Developing an Effective Media Strategy for Long-term Retention" for a discussion of these issues and suggestions on how to keep ahead of the eventual obsolescence of media, hardware, database and application software.)
- Decentralized decision making. Information may be moved to departmental applications and data warehouses for easier access by business units or for data analytics.

In all of these scenarios, records move along with non-records.

CHALLENGE: ROUTINE DATABASE PURGING

Database purge routines, the act of freeing up space or deleting data no longer needed for operations, are commonly based on the age or type of data. Purge routines can be set up in the application to run automatically or on a scheduled IT process. Because most transactional data only needs to be retained for a short period of time, purging old data can improve system performance and help IT manage server and storage costs. Before purge routines are set up or scheduled, consideration should be given to the database schema (which defines the relationships among data elements), the data dictionary and the organization's record retention schedule, otherwise compliance, business and legal requirements may be violated or records may be lost. Many databases have no purge capabilities without a significant amount of retro-fitting; organizations should adopt a "compliance by design" philosophy when building new systems to enable purge routines and other retention measures.

How to maintain record integrity?

When structured data is in motion, care must be taken to make sure that record structure and integrity are maintained. To ensure record integrity, records and information should be "authentic and unaltered" (source: ARMA). Records maintained in structured applications require that metadata be stored with the record and include, at a minimum:

- File names, creation date, owner and data location
- Security and signature requirements for application and end-user access
- Fields such as username or time stamp, which are populated automatically

STRATEGIES FOR LONG-TERM RETENTION AND DISPOSITION

Preserving long-term structured data records can be a straightforward task if data volume growth is modest and the production system can serve as both a repository for active information as well as an archive. But a "preserve in place" approach may not be practical as storage costs escalate and application performance suffers. Several alternative management strategies should be considered:

STRATEGY: BACKUP AND RESTORE

In this scenario, the production or archived database with index information and metadata are backed up regularly and restored when needed. This approach mitigates the potential loss of information due to purging, but may be overkill because all data is stored – not just a limited number of relevant, long-term records. In addition, it can be costly and time-consuming to recreate the database from the backup tapes. Finally, tape backup media is subject to degradation over time and it may be difficult to run the application on future generations of hardware. While a backup and restore strategy may be an attractive option for its apparent simplicity, in most cases it is better suited to meet disaster recovery and response requirements and not for long-term record retention.

STRATEGY: ARCHIVING

Structured data applications are archived by moving the content from a production system to a purpose-built archive or other offline storage area. Archiving strategies are most often implemented when immediate access and retrieval is not required in order to improve application performance or to reduce storage costs. Structured data archives are generally not suitable for preservation of long-term records, because searching, retrieving, reporting and properly disposing of records can be difficult, especially if data exists in multiple archive solutions/locations.

One way to avoid these problems is to build a "united view" of active and archive databases:

- Configure the active database consisting of recent transactions (e.g., up to three years) to be stored in "online" systems that support rapid and frequent access to data.
- Store the archive database with older transactions on higher-latency "near-line" storage.
- As long as the tables in each database are compatible (i.e., have the same number of columns and formats), a "union" query operation can be performed by an end-user or automated process, whereby data from the active and archive databases can be retrieved and presented together. There are several advantages to this method:
    - Production database size is reduced, improving application performance.
    - Usage and expenditure on costly, low-latency storage systems can be optimized.
    - The probable life expectancy of near-line storage is greater than that of production storage.
    - Archival data can be accessed more quickly than from offline storage.
    - Long- and short-term records can be retrieved with a single query.

STRATEGY: MIGRATE TO A DATA WAREHOUSE

Migrating structured data to a data warehouse application facilitates business unit access, as summary reports can be generated by end-users who run queries using data warehouse tools. This approach can be problematic for long-term retention as it may be difficult to access individual records and prove chain of custody unless all record metadata and index information is migrated so that reports produce only the desired results.

STRATEGY: EXTRACT AND SAVE IN A SUSTAINABLE FORMAT

With this approach, records are transformed into business objects in standard, natural-language formats such as .xml. There are several advantages to this strategy:

- Records are searchable and human-readable.
- Retrieval is simpler, as there is no need to run separate queries against multiple systems.
- Long-term access to information can be assured even as legacy applications are decommissioned.

This conversion effort usually requires custom coding and an understanding of the database schema, data dictionary and database purge processes in order to extract only

the required records and metadata from the original application. A document management system (DMS) or electronic content management (ECM) solution provides the functionality for applying governance and retention policy settings, search, extended metadata, chain of custody and disposition. Alternatively, storage as flat files on durable media or even printed out on paper may be appropriate.

CONCLUSION

Production applications that generate structured data are mission-critical for organizations. IT policies and procedures that focus on optimized performance and reduced downtime complicate the task of managing the retention and disposition of records that have long-term historical, legal or compliance retention requirements. The strategies outlined in this white paper, complemented by careful planning and cooperation between IT and RIM professionals along with business units they serve, means that the goals of each group are documented and understood. Implementing the right strategy ensures preserving the value of legacy data for end users, and satisfies compliance requirements while reducing IT support and maintenance costs.

ABOUT IRON MOUNTAIN

Iron Mountain Incorporated (NYSE: IRM) is a leading provider of storage and information management services. The company's real estate network of over 67 million square feet across more than 1,000 facilities in 36 countries allows it to serve customers around the world. And its solutions for records management, data management, document management, data center management and secure shredding help organizations to lower storage costs, comply with regulations, recover from disaster and better use their information. Founded in 1951, Iron Mountain stores and protects billions of information assets, including business documents, backup tapes, electronic files and medical data. Visit www.ironmountain.com for more information.