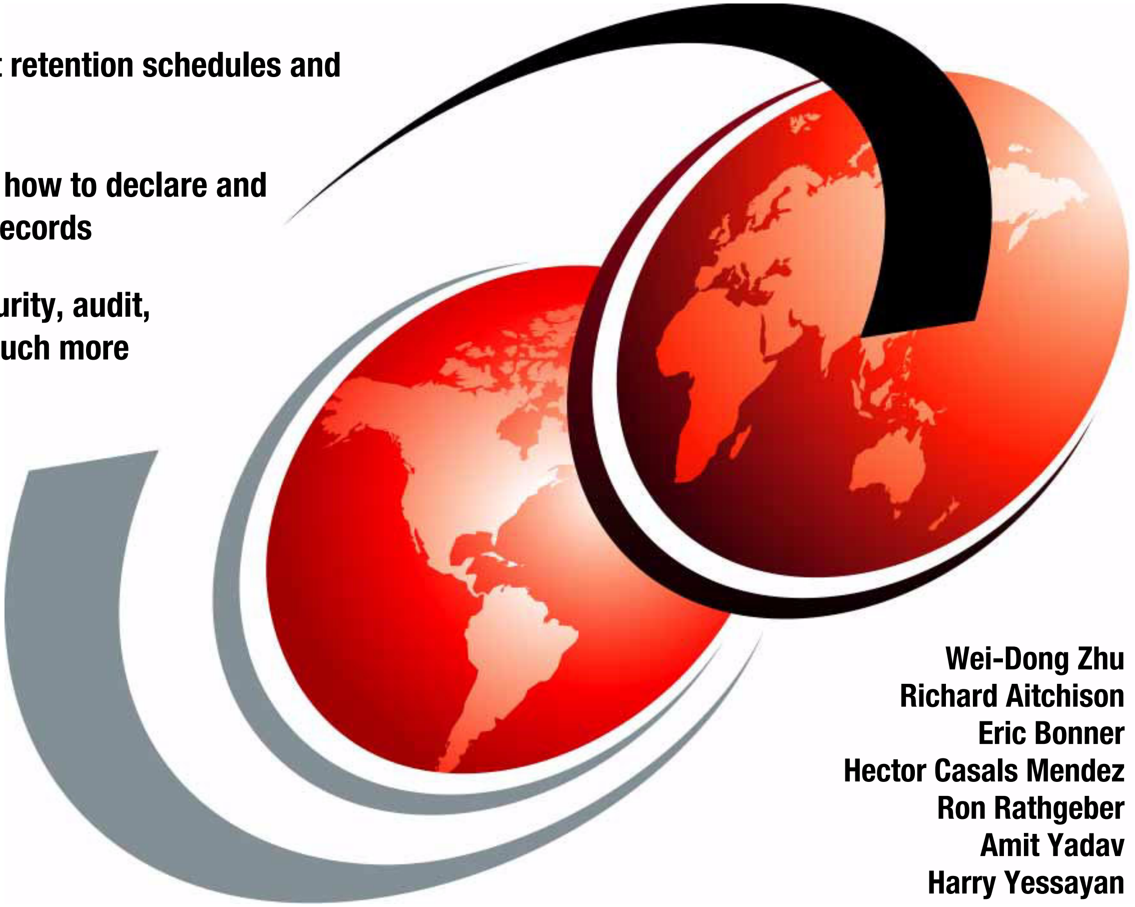
Understanding IBM FileNet Records Manager



- Learn about retention schedules and file plans
- Understand how to declare and dispose of records
- Review security, audit, hold, and much more

Wei-Dong Zhu
Richard Aitchison
Eric Bonner
Hector Casals Mendez
Ron Rathgeber
Amit Yadav
Harry Yessayan

Redbooks

ibm.com/redbooks

International Technical Support Organization

Understanding IBM FileNet Records Manager

May 2009

SG24-7623-00

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

First Edition (May 2009)

This edition applies to Version 4, Release 5 of IBM FileNet Records Manager (product number 5724-S19).

Contents

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

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

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

FileNet®
IBM®
Rational®
Redbooks®
Redbooks (logo) ®
WebSphere®

The following terms are trademarks of other companies:

FileNet, and the FileNet logo are registered trademarks of FileNet Corporation in the United States, other countries or both.

JBoss, and the Shadowman logo are trademarks or registered trademarks of Red Hat, Inc. in the U.S. and other countries.

EJB, J2EE, J2SE, Java, JDBC, JVM, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Visual Basic, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Pentium, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

Preface

Records management helps users address evolving governance mandates to meet regulatory, legal, and fiduciary requirements. Proactive adherence to information retention policies and procedures is a critical facet of any compliance strategy. IBM® FileNet® Records Manager helps organizations enforce centralized policy management for file plans, retention schedules, legal preservation holds, and auditing. IBM FileNet Records Manager enables your organization to securely capture, declare, classify, store, and dispose of electronic and physical records.

In this IBM Redbooks® publication, we introduce the records management concept and provide an overview of IBM FileNet Records Manager. We address records management topics, including the retention schedule, file plan, records ingestion and declaration, records disposition, records hold, and IBM FileNet Records Manager application programming interfaces (APIs). In addition, using a case study, we describe step-by-step instructions to implement a sample records management solution using IBM FileNet Records Manager. We provide concrete examples of how to perform tasks, such as file plan creation, records ingestion and declaration, records disposition, and records hold. The following company name appearing in this publication is fictitious: Fictional Insurance Company X. This name is used for instructional purposes only.

This book helps you to understand the records management concept, the features and capabilities that IBM FileNet Records Manager offers, and its usage.

The team that wrote this book

This book was produced by a team of specialists from around the world working at the International Technical Support Organization, Rochester Center.

Wei-Dong Zhu (Jackie) is an Enterprise Content Management Project Leader with the International Technical Support Organization. She has more than 10 years of software development experience in accounting, workflow processing, and digital media distribution. Jackie holds a Master of Science degree in Computer Science from the University of the Southern California. Jackie joined IBM in 1996. She is a Certified Solution Designer for IBM Content Manager and has managed and led the production of many Enterprise Content Management IBM Redbooks publications.

Richard Aitchison is an Enterprise Content Management PreSales Consultant with IBM Software Group in the United Kingdom. He has 15 years of experience in the field of enterprise content management, working in both presales and postsales roles. His areas of expertise include IBM document management and compliance solutions. Richard holds a degree in Computer Science from Hull University.

Eric Bonner is an Enterprise Content Management Presales Technical Solution Specialist with the IBM Worldwide ECM Proof of Technology team in the United States. He has over 20 years of experience in the software industry with an emphasis in software development, including Java™, Microsoft®, and Web technologies. Eric has 10 years of IBM FileNet product experience. He specializes in developing complex demonstrations and technical proofs of concepts involving IBM FileNet Content Management, Business Process Manager, Business Process Framework, eForms, Records Manager, EMail Manager, and Records Crawler. Eric has Bachelor of Science degrees in both Physics and Mathematics from University of California, Irvine.

Hector Casals Mendez is an Enterprise Content Management Presales Technical Solution Specialist with IBM Software Group based in Madrid, Spain. Hector has 18 years of Information Technology experience. Prior to joining IBM in 2007, Hector worked four years in FileNet Spain and six years for FileNet partners in South America and Spain. His areas of expertise include IBM FileNet Content Manager, Business Process Manager, eMail Manager, Records Crawler, eForms, and Web Site Manager. Hector holds a Masters Degree in Economic Cybernetics from Moscow, Russia.

Ron Rathgeber is an Enterprise Content Management Software Architect working on the IBM FileNet Records Manager product in the IBM Software Group, United States. Previously, he worked on the IBM FileNet Capture and FileNet WorkForce Desktop products. Ron has 20 years of experience in the imaging and content management fields, working in the software development group. He holds a Bachelor of Science degree in Information and Computer Science from the University of California, Irvine.

Amit Yadav is a Certified IBM FileNet Professional working with IBM Global Business Services as a Software Application Developer in India. He has three years of experience in IBM FileNet solution development. Amit holds a Bachelor of Technology (B.Tech.) degree in Computer Science. His areas of expertise include IBM FileNet Workplace, Business Process Manager, and Records Manager development and customization.

Harry Yessayan is a Senior Systems Architect for IBM Enterprise Content Management Lab Services in the United States. He has more than 10 years of IBM FileNet Content Manager and Business Process Manager product experience and 20 years of combined industry and academic software development, research, and consulting experience. Harry currently specializes in providing customers with detailed requirements analysis and technical configuration designs to help them implement and configure IBM FileNet Records Manager systems for compliance solutions. In addition, he has experience in product training, specializing in IBM FileNet APIs for custom application development. Harry holds an Master of Science degree in Information and Computer Science from University of California, Irvine.

Thanks to the following people for their contributions to this project:

Martin Shramo
Frank Ayars
Edwards (Ned) Bader
Jay Devaney, Jr.
Stewart Imagawa
James Jin
Lijing Lin
Lizing Lin
Yi-Jen Lin
Chi Nguyen
Michelle Oliver
Vincent Pham
Lannie Truong
Jeff Wallace
Li Zhou
IBM Software Group, Costa Mesa, California

Become a published author

Join us for a two-week to six-week residency program. Help write a book dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You will have the opportunity to team with IBM technical professionals, IBM Business Partners, and Customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you will develop a network of contacts in IBM development labs and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us.

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- Use the online **Contact us** review IBM Redbooks publications form found at:

  ibm.com/redbooks

- Send your comments in an e-mail to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Part 1

Concept

In this part, we introduce the records management concept and provide an overview of IBM FileNet Records Manager, the product. We address records management topics, including the retention schedule, file plan, records ingestion and declaration, records disposition, records hold, and IBM FileNet Records Manager application programming interfaces (APIs).

1

Introduction to records management

In this chapter, we provide an introduction to help you understand records management. We describe the need for records management, key concepts, and the implementation methodology.

We cover the following topics in this chapter:

- Records
- Records management
- Legal considerations
- Regulatory requirements
- Enterprise-wide records management solution methodology
- Records management standards and guidelines
- Records management key concepts

1.1 Records

A *record* is any type of content stating results achieved, pertaining to, and providing evidence of activities performed. There are four essential characteristics of a record. They are:

- *Authenticity*

  A record must be what it purports to be.

- *Reliability*

  A record must be a full and accurate representation of the transactions, activities, or facts to which it attests.

- *Integrity*

  A record must be complete and unaltered.

- *Usability*

  A record must be able to be located, retrieved, presented, and interpreted.

A record:

- Must have fixed content
- Provides evidence of a transaction, activity, or fact that has legal or business value
- Has a specific retention period that is based on the business policy and regulatory rules
- Is owned by a company, enterprise, or government

A record is generally retained for analysis or historical purposes as a representation of what occurred. Records can be trade instructions, trade confirmations, articles of incorporation, bylaws, or standard operating procedures.

Records can be in any format and can take the form of paper records, microfiche, electronic documents, e-mail, fax, instant messaging, collaboration content, voice recording, wireless communication content, audio, video, shared drive content, and Web content. E-mail systems are common sources of records as a history of the discussions and the decisions of a company, which is a primary reason for eDiscovery on e-mail today.

Records can reside in any medium, such as diskette, tape, optical disks, and shared drives. Records can be generated internally in a company or can be received from other sources.

Records are similar to other assets of a company. They are valuable and subject to government regulations. Many countries have legislation about record retention. Most laws are applicable to physical and electronic records. Certain laws specify an active and inactive retention period, and specific laws have a special compliance requirement for storage media. For example, paragraph (f)(2)(ii)(A) of the U.S. Securities and Exchange Commission (SEC) Rule 17a-4 requires electronic storage media to preserve the records of broker-dealers exclusively in a non-rewritable and non-erasable format.

Note: A record differs from a document. A *document* can be modified whereas a record cannot be modified. Record versioning is not allowed. However, you can have multiple versions of a document, where each version is considered a record.

A record and its complete end-to-end history are vital and important for compliance and defensible eDiscovery. Having and using the compressive auditing and security capabilities in IBM FileNet Records Manager help preserve and provide end-to-end records lifecycle information at every step across the entire records collection process. We provide the product highlights, its capabilities, and the key benefits in Chapter 2, “IBM FileNet Records Manager system and architecture” on page 29.

1.2 Records management

Often in a corporate environment, documents are created or captured in a decentralized environment with no overall oversight. Documents are named and filed according to the individual’s preferences and often are subject to duplication. Documents might be kept for too long, which can lead to increased storage cost. If litigated, companies must spend resources to locate documents (eDiscovery). The best practice is being able to dispose of records according to an auditable schedule, thus reducing the eDiscovery scope and production costs. In certain cases, companies cannot locate the documents or cannot locate them within the required time, which can lead to financial penalty, sanctions, or worse, *damage* to a company’s reputation.

The International Organization for Standardization (ISO) 15489[1] 2001 standard defines records management as “*The field of management responsible for the efficient and systematic control of the creation, receipt, maintenance, use and disposition of records, including the processes for capturing and maintaining evidence of and information about business activities and transactions in the form of records.*”

[1] ISO official Web site: http://www.iso.org/iso/home.htm

Records management is a formal and structured process of identifying record information, preserving needed content, and destroying content that is no longer needed. Destruction of a record is allowed only after the approved retention period has been reached.

Figure 1-1 on page 6 provides a records management overview.

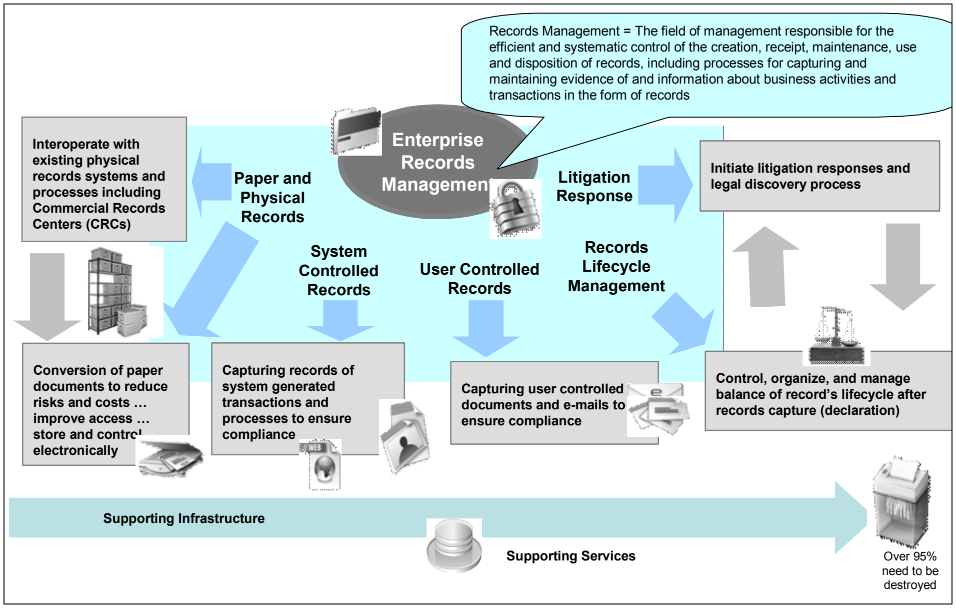

Records Management = The field of management responsible for the efficient and systematic control of the creation, receipt, maintenance, use and disposition of records, including processes for capturing and maintaining evidence of and information about business activities and transactions in the form of records
Enterprise Records Management
Interoperate with existing physical records systems and processes including Commercial Records Centers (CRCs)
Paper and Physical Records
Litigation Response
Initiate litigation responses and legal discovery process
System Controlled Records
User Controlled Records
Records Lifecycle Management
Conversion of paper documents to reduce risks and costs … improve access … store and control electronically
Capturing records of system generated transactions and processes to ensure compliance
Capturing user controlled documents and e-mails to ensure compliance
Control, organize, and manage balance of record's lifecycle after records capture (declaration)
Supporting Infrastructure
Supporting Services
Over 95% need to be destroyed

Figure 1-1 Records management overview

Records management involves:

- Identifying the information that needs to be declared as a record
- Categorizing the records
- Retaining records for a specific period of time
- Disposing of records when an organization is no longer obligated to retain them
- Preserving an audit trail of all activities that are related to the records

There are three key factors in records management:

- Records identification: Determine what organizational information represents records, where the records are located, and how to classify them.
- Records preservation: Ensure that records are maintained and unalterable to avoid spoilation. Ensure that records are accessible until the appropriate retention period has elapsed.

- Records disposition: Ensure that records are disposed after the required retention period has ended. Disposition is the final phase of a record's lifecycle. Not all records are destroyed at the end of the lifecycle. Disposition, therefore, can include, but is not synonymous with, records destruction.

Important: Because records might be required to comply with government regulations or to protect a company from liability, *the company* controls the records, not the users or the creators of the records.

Records management differs from content management. *Content management* provides the ability to capture, store, and manage content. *Records management* works within this infrastructure to apply formal, rules-based management of the retention and disposition of the stored content.

Records management is media-independent. Physical and electronic record types or content can be managed together (sometimes referred to as "*hybrid records*") or separately with distinct retention schedules and file plans. Usually, a record is not one document or content item, but it is made up of a collection of related documents, along with key metadata and audit history.

An effective records management solution manages the lifecycle of corporate records from creation to disposition. Figure 1-2 illustrates the complete enterprise content management lifecycle process. Content, such as images, e-mails, documents, and transactions are first created. The content is then captured and classified into a file plan. Records go through the records lifecycle. At the end of a record's lifecycle, records disposition occurs: the records are either expunged, shredded, or permanently archived. Not all records are destroyed at the end of their lifecycle.

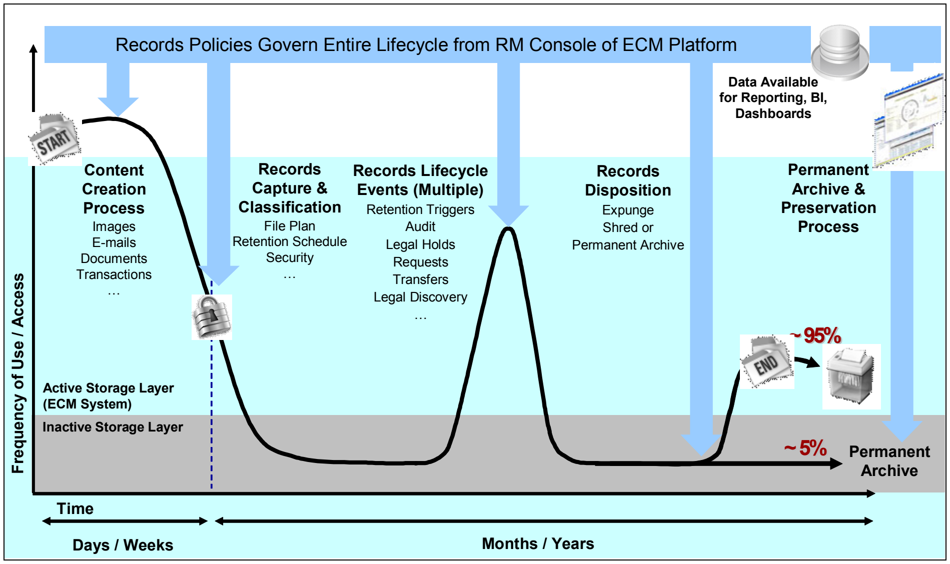


Figure 1-2 Enterprise records management lifecycle process

Typically, during the document creation process, a number of revisions of the document are created over a period of weeks or months. When the document is finally completed, a decision needs to be made, either by a rule that is defined within a workflow or by a user manually, whether the document qualifies as a business record. If yes, the document becomes an official record, after which it can no longer be altered and is subject to *retention rules*. Ideally, users need to follow standardized rules for records when they first create the items.

At the time of disposition, an authorized person can archive or expunge the record.

> **Note:** *Expunge* is a term that implies irrevocably deleting the records so that even document forensics (using, for example, low-level file undelete) cannot recover any aspect of the records. Department of Defense (DoD) 5015.2 tests perform this type of low-level validation. You expunge records when you execute *destroy* as an records disposition option. Expunging records is an expensive operation and is typically optional. In certain cases, the expunge is not required by the business case and might not be configured. In a DoD scenario[a], expunge is required.

a. Our use of DoD means the current DoD 5015.02 V3 Records Management Certification standard issued by Joint Interoperability Test Center (JITC) DoD.

Records management involves retaining corporate records for the minimum appropriate retention period to meet business and regulatory requirements (laws, policies, and regulations about the business). The requirements usually define the minimum duration for which to keep records. Best practices and other business needs can determine a retention period that is longer than the base duration that is set in regulations.

Records management also implies the management of the full records lifecycle: providing the guidelines for appropriate records creation, providing that information to those people who need to know it, retaining the records, and securely disposing of the records based on their disposition schedules.

Records management involves managing risks and costs for retaining corporate records. Companies must be able to demonstrate or prove that they have a records retention policy and procedures in place and that they enforce the policy and procedures consistently.

The benefits of records management include:

- *Meeting compliance and litigation requirements*: Industry and government regulations often impose various requirements for records. Timely destruction of records in full compliance reduces the risk of exposure in case of litigation.
- *Safeguarding records* for business audit or business continuity reasons. Records are vulnerable to natural disaster, accidents, theft, or mishandling. An efficient records management solution helps to identify and protect against these threats, which is especially important for vital records that are essential to the continued operation of a company.
- *Meeting fiscal requirements*: Ensure that companies comply to their fiscal requirements for record retention.
- *Ensuring operational efficiency*: Ensure that corporate information is captured, retained, and disposed properly.
- *Containing cost*: Ensure that records are destroyed after their required retention period, which can reduce storage costs and space requirements.

Businesses need a holistic approach to records management throughout a record’s lifecycle from capture to creation, maintenance and usage, protection and preservation, to its final disposition. Companies need to be prepared to prove the authenticity of the records, the trustworthiness of the processes, and the integrity of the records management systems. Strong accountability through records management provides integrity and authenticity, to prove compliance, especially during an audit.

Note: The key objectives of records management include risk mitigation, compliance, and cost containment of record keeping. A blanket retention policy of keeping everything indefinitely is no longer the best practice.

1.3 Legal considerations

Regulatory bodies and governments impose rules dictating various retention periods on various *record series*. Compliance specifies what documents a company needs to preserve and how the documents are preserved to comply with laws and regulations. One of the objectives of an effective records management program is to preserve records for the appropriate length of time.

Note: *Record series* is a records management term that means a group of related records that can be filed as a unit for retention purposes.

Companies that destroy records prior to their legal retention period incur massive financial penalties in the case of litigation. Alternatively, retaining records after the required retention period can cause an equally large exposure for a company, due to potentially incriminating documents being discoverable during a litigation action. A secondary consideration of not disposing of records in a timely and legal manner is the increased storage requirements and costs.

Note: *Spoliation* generally refers to the intentional destruction of information relevant to a legal matter in which an organization is involved. Organizations must ensure that they do not alter the original format or context of the records, and that they can also deliver them in a provable, auditable original or representative form.

When there is a court or regulatory order, companies need to go through a *legal discovery process*. This process often requires the companies to search across all documents whether they are records or not and identify those documents that match the discovery order. Any document in any medium that has information relevant to the subject matter is potentially discoverable and needs to be preserved for as long as the lawsuit is anticipated or pending. These records and documents need to be placed on *hold* so that the normal retention schedule and disposition processing do not proceed until the hold is removed.

Note: A *hold* is also known as a *legal hold*, which is an action taken on record collections to ensure that they are not destroyed as part of their normal retention schedule life and are kept accessible beyond their scheduled date of destruction. Records under legal hold are protected from any possible destruction until the hold is lifted. A legal hold is usually driven by legal discovery litigation needs. There might also be other types of holds, such as audit holds and investigatory holds.

For DoD, the term *Freeze* is used instead of hold.

Organizations can also conduct a discovery process for the purposes of due diligence, internal investigations, and other reasons.

Courts are beginning to expect companies to have systems in place to speed up the act of discovery. The new Federal Rules of Civil Procedure has guidelines where a company can claim undue burden and the company can narrow the discovery request to precisely the information that is relevant to the matter. If the companies cannot produce relevant documentation in a timely manner, they can incur considerable financial penalties or damage to the company's reputation.

1.4 Regulatory requirements

Companies need to have a good understanding of applicable regulations, how to identify records, and how to identify the corresponding retention requirements on the records. The complexity of managing records is increased by evolving compliance rules and regulations as illustrated in Figure 1-3.

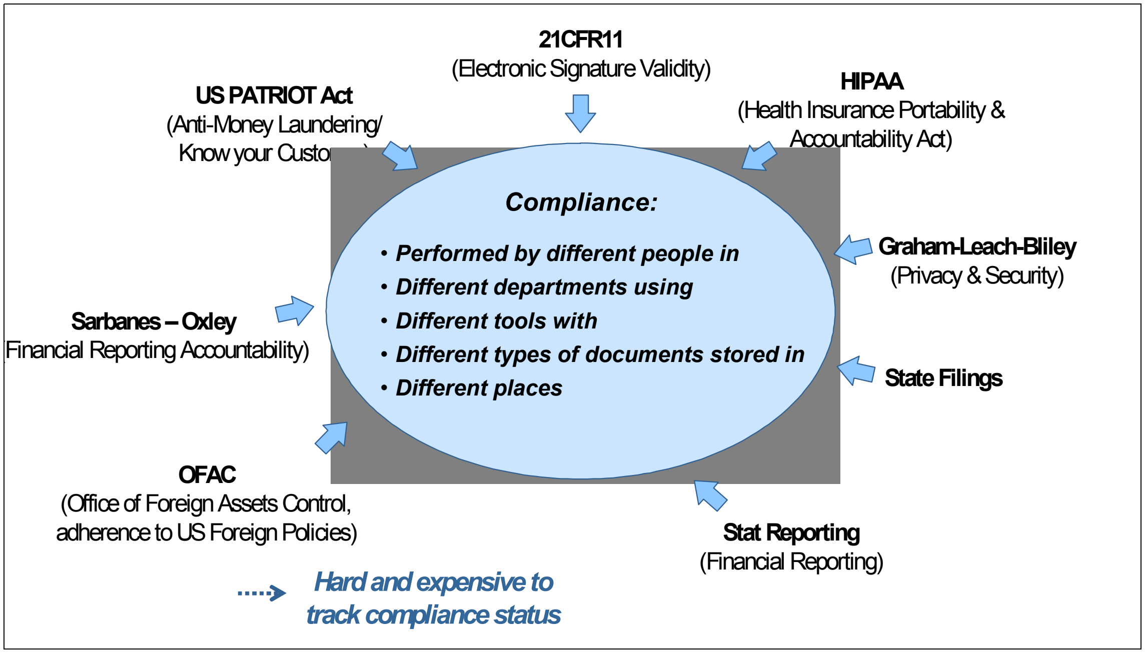


Figure 1-3 Examples of U.S. regulations pertinent to compliance

Compliance is the act of adhering to and demonstrating adherence to internal or external regulations. A *regulation* is issued by either a federal or state agency as directed by legislation. It is a compromise between prohibition and no control at all. For example, the sale and consumption of prescription drugs are controlled by regulations as are other areas, such as the financial sector.

Compliance with regulations and laws means:

- Interpreting what the regulations and laws say
- Understanding where your company currently stands
- Documenting a plan for achieving compliance
- Executing the plan

- Devising measures and controls to prove that your company has implemented the plan

Addressing regulatory requirements is not a straightforward exercise due to the complexity of legislation. Certain U.S. pertinent legislation or requirements requiring financial institutions to secure records include:

- Investment Advisors Act

  The Investment Advisors Act rule 204-2 establishes record keeping requirements for books and records to be maintained by investment advisers.

- Securities and Exchange Commission (SEC)

  The SEC Act of 1934 for broker-dealers and transfer agents section 17a requires securities brokers, dealers, investment companies, financial advisers, and transfer agents to keep records of interoffice communications and communications with customers:

  - Section 17a-3 requires that all members of a national securities exchange, including all brokers and dealers, must keep current a variety of books and records that relate to their businesses.
  - Section 17a-4 requires that certain records retained by brokers and dealers must be preserved for at least six years, the first two years in an easily accessible place, while other records must be retained for at least three years, the first two years in an easily accessible place.

- New York Stock Exchange (NYSE)

  The NYSE rule 440 requires brokers and dealers to make and preserve books and records as prescribed by the NYSE.

- Bank Secrecy Act (Anti-money laundering statutes and rules)

  The Bank Secrecy Act requires businesses to keep records and file reports that are determined to have a high degree of usefulness in criminal, tax, and regulatory matters. Agencies use these documents to identify, detect, and deter money laundering whether it is in furtherance of a criminal enterprise, terrorism, tax evasion, or other unlawful activity. Businesses must report cash payments of over $10 000 received in trade or business from one buyer as a result of a single transaction or as a result of two or more related transactions.[2]

- State statutes

  State or local laws also govern the requirements of record keeping. Each state has its own jurisdiction.

[2] Information taken from the Internal Revenue Service Web site at: http://www.irs.gov/businesses/small/article/0,,id=152532,00.html

- Sarbanes-Oxley Act (SOX)

  The Sarbanes-Oxley Act requires that firms that audit companies governed by the SEC retain all relevant documentation to protect mishandling of information.

- Gramm-Leach-Bliley Act

  The Gramm-Leach-Bliley Act is a U.S. Federal law enacted to control ways that financial institutions handle private information for individuals.

- The Office of Foreign Assets Control

  The Office of Foreign Assets Control (OFAC) of the U.S. Department of the Treasury administers and enforces economic and trade sanctions based on U.S. foreign policy and national security goals against targeted foreign countries, terrorists, international narcotics traffickers, and those individuals engaged in activities related to the proliferation of weapons of mass destruction.

1.5 Enterprise-wide records management solution methodology

Companies require a holistic records management program to meet today's compliance and business needs. To ensure a successful implementation of an enterprise records management solution, we recommend the following methodology as shown in Figure 1-4 on page 14:

- Obtain corporate sponsorship and stakeholders' buy-in.
- Assess and evaluate the company's current situation and identify gaps.
- Gather business and technical requirements.
- Engineer business processes and deploy technology.
- Review and monitor (audit) the processes continuously.

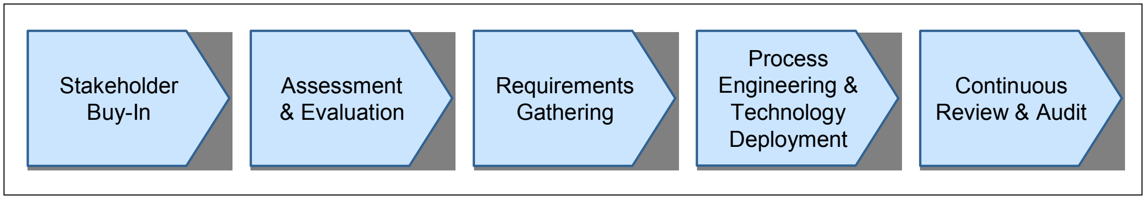

Stakeholder Buy-In
Assessment & Evaluation
Requirements Gathering
Process Engineering & Technology Deployment
Continuous Review & Audit

Figure 1-4 Methodology to implement an enterprise-wide records management program

1.5.1 Obtaining corporate sponsorship and stakeholder buy-in

A successful records management program requires corporate governance from the top down as well as enforcement throughout the company. Executive sponsorship is key to the success of an enterprise deployment. Other stakeholders include (but are not limited to) Records Managers, Office of General Counsel, compliance officers, and a cross-functional team that includes representatives from all business areas.

Note: The *Office of General Counsel* provides legal and policy advice within the company.

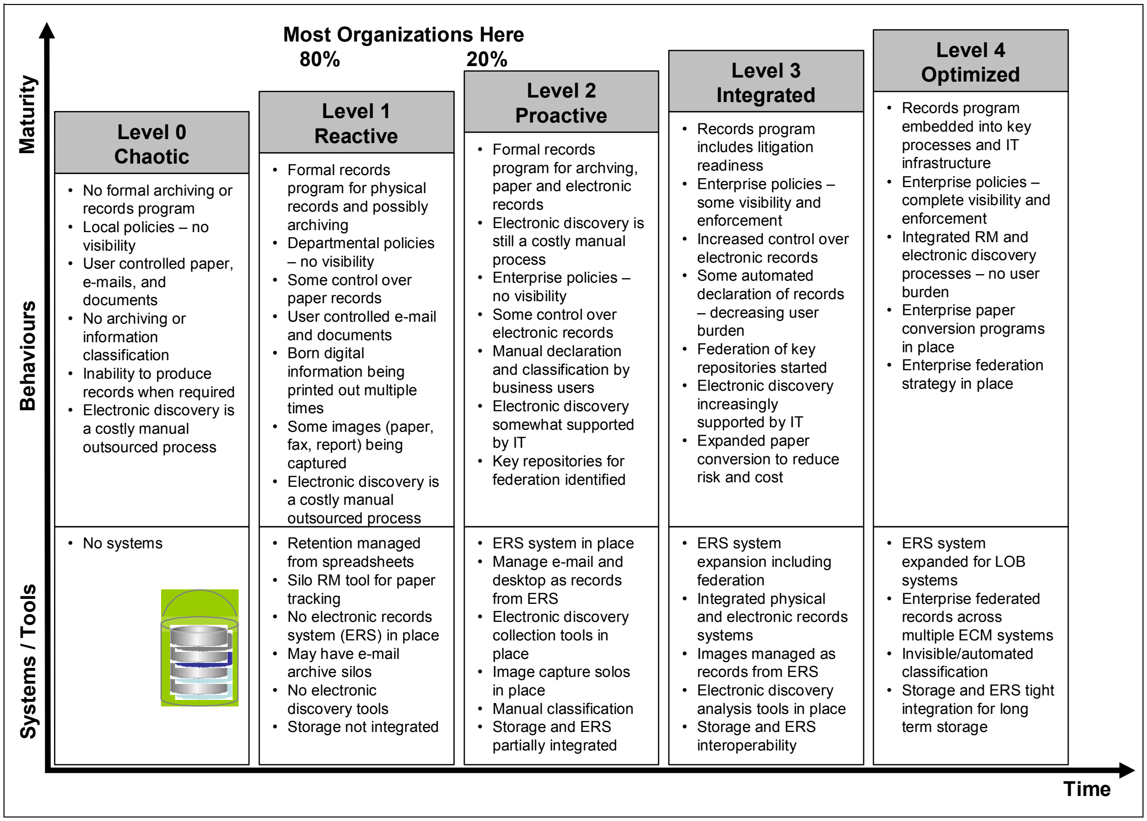
A records management program usually requires significant funding. Failure to do so can be even more costly in the case of litigation. It is important to get an executive sponsor and to engage across all functional teams for enterprise-wide participation.

1.5.2 Assessing and evaluating current policies and procedures

During the assessment and evaluation phase, the company’s current situation is reviewed. Assess the company’s assets, including the company’s retention policies and procedures and retention schedules. Records retention procedures must reflect the company’s records retention policies. The outcome of this exercise is to identify requirements and gaps and to establish priorities.

Using an objective compliance and discovery maturity model

One way to assess the health of a company's current records retention and management practices is to use an objective compliance and discovery maturity model as shown in Figure 1-5.



	Level 0 Chaotic	Level 1 Reactive	Level 2 Proactive	Level 3 Integrated	Level 4 Optimized
Behaviours	• No formal archiving or records program • Local policies – no visibility • User controlled paper, e-mails, and documents • No archiving or information classification • Inability to produce records when required • Electronic discovery is a costly manual outsourced process	• Formal records program for physical records and possibly archiving • Departmental policies – no visibility • Some control over paper records • User controlled e-mail and documents • Born digital information being printed out multiple times • Some images (paper, fax, report) being captured • Electronic discovery is a costly manual outsourced process	• Formal records program for archving, paper and electronic records • Electronic discovery is still a costly manual process • Enterprise policies – no visibility • Some control over electronic records • Manual declaration and classification by business users • Electronic discovery somewhat supported by IT • Key repositories for federation identified	• Records program includes litigation readiness • Enterprise policies – some visibility and enforcement • Increased control over electronic records • Some automated declaration of records – decreasing user burden • Federation of key repositories started • Electronic discovery increasingly supported by IT • Expanded paper conversion to reduce risk and cost	• Records program embedded into key processes and IT infrastructure • Enterprise policies – complete visibility and enforcement • Integrated RM and electronic discovery processes – no user burden • Enterprise paper conversion programs in place • Enterprise federation strategy in place
Systems / Tools	• No systems	• Retention managed from spreadsheets • Silo RM tool for paper tracking • No electronic records system (ERS) in place • May have e-mail archive silos • No electronic discovery tools • Storage not integrated	• ERS system in place • Manage e-mail and desktop as records from ERS • Electronic discovery collection tools in place • Image capture solos in place • Manual classification • Storage and ERS partially integrated	• ERS system expansion including federation • Integrated physical and electronic records systems • Images managed as records from ERS • Electronic discovery analysis tools in place • Storage and ERS interoperability	• ERS system expanded for LOB systems • Enterprise federated records across multiple ECM systems • Invisible/automated classification • Storage and ERS tight integration for long term storage

Figure 1-5 Compliance and discovery maturity model

This model enables companies to baseline business records retention and management practices using an objective matrix:

- Level 0 - Chaotic

  Companies at this level exhibit the following behaviors:

  – No formal archiving or records program
  – Local policies - no visibility
  – User-controlled paper, e-mails, and documents
  – No archiving or information classification
  – Inability to produce records when required

There is no system in place. Electronic discovery for companies at this level is a costly manual outsourced process.

- Level 1 - Reactive

  Companies that are at level 1 exhibit the following behaviors:

  – Formal records program for physical records and possibly archiving
  – Departmental policies – no visibility
  – Some control over paper records
  – User-controlled e-mail and documents
  – Born digital information being printed multiple times
  – Some images (paper, fax, and reports) being captured

  At this level, retention is managed from spreadsheets. Companies also use a records management tool for paper tracking. There is no electronic records system (ERS) in place. The company might have e-mail archive. There are definitely no electronic discovery tools. Storage is not integrated.

  Electronic discovery for companies at this level is a costly manual outsourced process.

- Level 2 - Proactive

  Companies that are at level 2 exhibit the following behaviors:

  – Formal records program for archiving, paper, and electronic records
  – Electronic discovery is still a costly manual process
  – Enterprise policies – no visibility
  – Some control over electronic records
  – Manual declaration and classification by business users
  – Electronic discovery somewhat supported by IT
  – Key repositories for federation identified

  Companies have ERS system in place. They manage e-mail and desktop as records from ERS. There are a electronic discovery collection tools in place. Image capture is in place. Classification is still manual. Storage and ERS are partially integrated.

- Level 3 - Integrated

  Companies at this level exhibit the following behaviors:

  – Records program, including litigation readiness
  – Enterprise policies – some visibility and enforcement
  – Increased control over electronic records
  – Some automated declaration of records – decreasing user burden
  – Federation of key repositories started
  – Electronic discovery increasingly supported by IT
  – Expanded paper conversion to reduce risk and cost

Companies achieving this level of maturity model have an ERS system to include federation. They also have integrated physical and electronic records systems. Images are managed as records from ERS. Electronic discovery analysis tools are in place. Storage and ERS are interoperable.

- Level 4 - Optimized

  Companies at this level exhibit the following behavior:

  - Records program embedded into key processes and IT infrastructure
  - Enterprise policies – complete visibility and enforcement
  - Integrated RM and electronic discovery processes – no user burden
  - Enterprise paper conversion programs in place
  - Enterprise federation strategy in place

  Companies achieving this level of maturity model have ERS system expanded for line of business (LOB) systems. Enterprise federated records are across multiple ECM systems. Classifications are automated and are invisible to users. Storage and ERS are tightly integrated for long-term storage.

Road map for records management solution implementation

After assessing the company for its current records management maturity and identifying a desired end-state, develop and document a road map to achieve compliance. Execute the road map by engineering business processes and deploying the records management technology.

Figure 1-6 is an example of a high-level graphical representation of a road map. Policy, procedure, and technology are the keys to a successful enterprise records management implementation. During the design process, research, assess, and account for important business requirements.

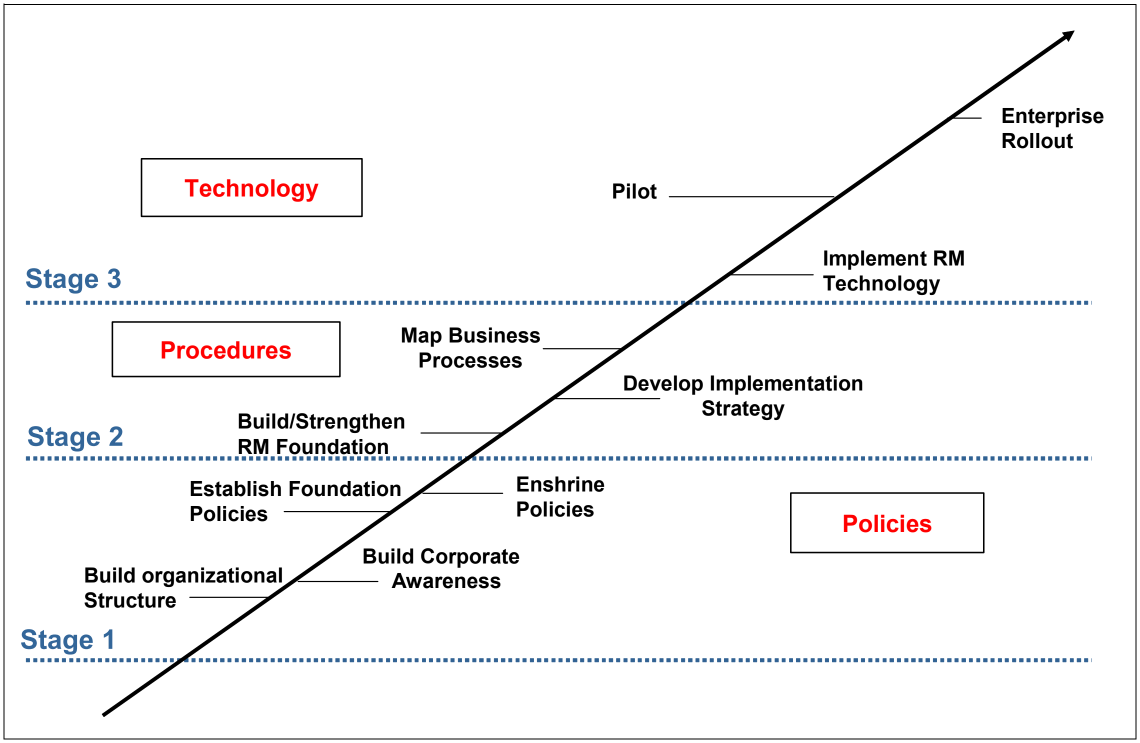

Enterprise Rollout
Technology
Pilot
Implement RM Technology
Stage 3
Procedures
Map Business Processes
Develop Implementation Strategy
Build/Strengthen RM Foundation
Stage 2
Establish Foundation Policies
Enshrine Policies
Policies
Build Corporate Awareness
Build organizational Structure
Stage 1

Figure 1-6 Records management solution implementation time line: Example of a road map

Organizational readiness

Figure 1-7 is a visual representation of an output of the assessment activities that measure the readiness of records management within a company.

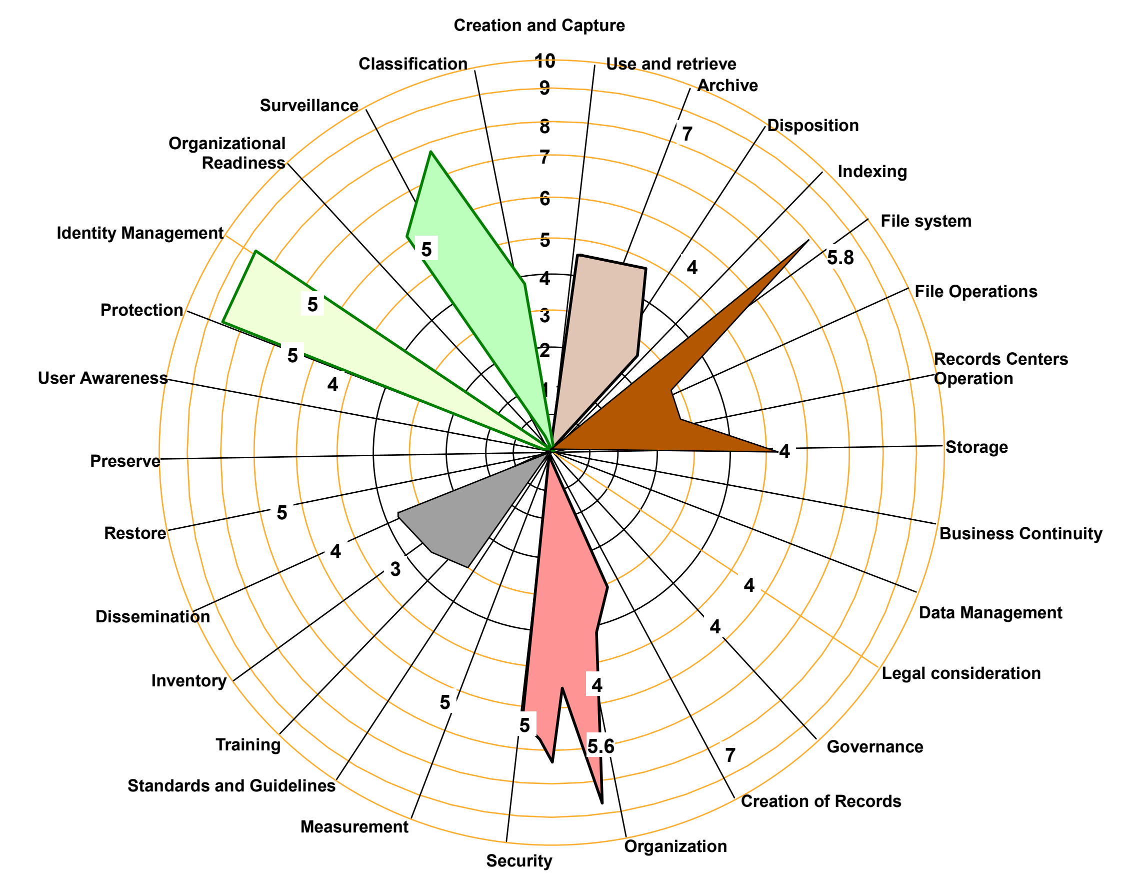


Figure 1-7 Example of an artifact from assessment

The guidelines for this assessment include:

1. Identify the key measurement areas for records management.
2. Identify the departments that receive records management services. Conduct an interview with members of selected departments.
3. Review the results from the interviews and address specific issues or concerns that are raised by the departments.
4. Analyze the results of the surveys and determine the departmental rating for each key measurement area.

5. Compare the departmental satisfaction rating with the target. Determine the course of action needed to bring about improvement and track those actions.

This type of visual representation can serve as a guideline against which the company can assess every aspect of records management lifecycle maturity and other areas to be measured. This assessment provides a relative strength of the area against the objective maturity model as shown in the previous section. It identifies the gap and brings attention to the area that requires improvement to achieve the desired end-state. The next step in the methodology is to identify the gaps and to determine the business and technology requirements.

1.5.3 Gathering business and technical requirements

Part of the implementation process is gathering business and technical requirements.

Business requirements

Through the analysis of existing documentation, the engagement with the stakeholders, and the interviews with individuals of various business areas, develop future requirements. Gather business requirements from business users. Understand the business needs and then derive the technical implementation from the business requirements.

The business requirement gathering steps can include the following activities (the list is not meant to be exhaustive):

- Confirm participants in the requirements collection
- Interview and engage business resources in the requirements validation process:
  - Develop interview guide
  - Develop schedule
  - Complete interviews and surveys
- Develop and prioritize mandatory, preferred, and optional requirements

Technical requirements

A records management application is a tool to help solve a business need that often involves process changes. Standards for records management are emerging and evolving globally. These standards provide a *baseline* for the technical requirements. Specific worldwide records management standards include DoD 5015.2, TNA, DOMEA, VERS, ISO 15489, and MoReq. We discuss these standards in 1.6, “Records management standards and guidelines” on page 22.

1.5.4 Engineering business processes and deploying technology

Based on the business and technical requirements, select a records management solution that best fits your company’s need. Together with the deployment of technology, streamline or automate the business processes. Technology is just one piece to the overall puzzle. User training and consistently applied policies are key to the success of an effective records management solution.

1.5.5 Reviewing and monitoring the processes

When a program is in place, employees need to comply with the records management policies and procedures. Review the new policies and procedures regularly to ensure regulatory compliance and modify the processes as needed. A successful records management program is a continuous improvement process.

1.6 Records management standards and guidelines

Records management standards are the guidelines to manage records defined by government agencies in various countries. In this section, we present several of the major standards.

> **Note:** A *records management standard* is defined by a particular authority or government for its own requirements. It might be applicable for other organizations. Therefore, an organization must assess its requirements before adopting any standard.

ISO 15489 information and documentation: Records management

The ISO 15489 standard is recognized worldwide as establishing the baseline for excellence in records management programs and implementing records management software applications. It is a process standard that provides a blueprint for the establishment, structure, monitoring, and auditing of a best practice records management program and software applications. It allows an organization to efficiently and effectively record and retrieve information, thus enhancing decision-making, productivity, and accountability, and at the same time, reducing the risk of exposing information.

This standard does not focus on records management technology solutions, but it encompasses all aspects of a records management program and software applications. There is therefore no software certification program for ISO 15489. If an organization implements an ERMS, this implementation is considered an enabler for ISO 15489.[3]

U.S. Department of Defense (DoD) 5015.2

The United States Department of Defense (DoD) Design Criteria Standard for Electronic Records Management Software Applications, better known as DoD 5015.2, debuted in 1997. Since then, it has become a common standard for U.S. government agencies, including the National Archives and Records Administration (NARA). It provides a formal certification program that private sector businesses routinely use as a way to evaluate or short-list records management software for potential purchases. DoD 5015.2 is also the starting point for base use cases for the retired UK National Archive (TNA) and the new European MoReq2 standards.

In June 2002, Chapter Four was added to the specification with additional requirements for records management applications, supporting classified (for instance, top secret, secret, and confidential) records, expanded audit requirements, more user-defined metadata fields, and guidance about e-mail record support.

A third revision of the specification came out in 2007. Version 3 adds:

- Requirements for interoperability between records management systems, export and import capabilities, and accession to NARA
- Privacy Act and Freedom of Information Act (FOIA) considerations (optional requirements)

> **Note:** *Accession* is physically archiving and transferring records from one records management system to another records-holding authority. It is one type of record disposition options.

MoReq

MoReq is a European specification that describes Model Requirements for the Management of Electronic Records (MoReq). It focuses mainly on the functional requirements for the management of electronic records by an Electronic Records Management System (ERMS). This specification is written to be equally

[3] ISO official Web site: http://www.iso.org/iso/home.htm

applicable to public and private sector organizations that want to introduce ERMS or that want to assess the ERMS capability that they currently have in place.

MoReq2

MoReq2 is the next generation of the MoReq standard. MoReq2 was formally published in March 2008. MoReq2 provides testing schemes, a feature that was not available in the MoReq. It has also taken input from newer records management standards and best practices, provides a software certification testing program for vendors, and because it is a European standard, it now provides the ability for different countries to have localized variations:

> *"However, information technology has changed since 2001. There have been growth and evolutionary change in many technology areas that affect the creation, capture and management of electronic records. This new version of MoReq, called MoReq2, addresses the impacts of that technological change. It also takes into account new standards and best practices that have been developed over the last several years. Accordingly, it is written as an evolutionary update of the original MoReq.*
>
> *MoReq2 for the first time also allows for a software testing regime to be implemented. It is written specifically to support the execution of independent compliance testing and a suite of compliance tests has been developed and published in parallel with the model requirements themselves. The need for rigorously worded, testable requirements has led to many changes of wording and expression in MoReq2.*
>
> *Finally, the years of experience in using and applying MoReq have pointed out the need for national variations to take into account different national languages, legislation, regulations, and record keeping traditions. For this reason, MoReq2 introduces for the first time a moderated mechanism – called "chapter zero" – to allow member states to add their unique national requirements."*[4]

Victorian Electronic Records Strategy (VERS)

VERS is an Australian standard developed by Public Record Office Victoria (PROV) to provide guidelines for capturing, managing, and preserving electronic records in the state of Victoria.[5] While the other standards mentioned here really focus on requirements for a records management solution, VERS concentrates

[4] *MODEL REQUIREMENTS FOR THE MANAGEMENT OF ELECTRONIC RECORDS, UPDATE AND EXTENSION, 2008, MoReq2 SPECIFICATION*, © European Communities, 2008. http://ec.europa.eu/transparency/archival_policy/moreq/index_en.htm

[5] Public Record Office Victoria, Australia, Victorian Electronic Records Strategy: http://www.prov.vic.gov.au/vers/standard

on defining a standard for the long-term preservation of electronic records. It is trying to ensure that an electronic record created today using current technology can:

- Be viewable in 10, 20, 50, or 100 years. The problem exists today. It is getting increasingly difficult to try to view a document that was created by a word processor 15 years ago, because current vendors drop support for these older formats. We need long-term digital preservation (LTDP).
- Have context, so that it is understood exactly from where the record came, who the author was, and to what it is related.

1.7 Records management key concepts

In this section, we provide records management key concepts. They are critical to a successful records management process.

File plan

A *file plan* is a structured, functional-based or organizational-based filing schema that a records management system uses to support a retention schedule. There is no universal file plan for all companies. Each file plan is unique and depends upon the types of businesses with which the company deals. A file plan specifies how records are organized hierarchically in a records management environment.

File plan construction is critical to an efficient and successful records management system. A file plan generally determines the security and disposition of records. By default, child entities inherit the security and disposition schedule of their parent container.

File plans for a company must be constructed only after analyzing a company's business and functions in-depth. A best practice for file plan creation is by Function/Activity/Transaction where *Function* is the major work performed by an organization, *Activity* might be a department, and *Transaction* might not equal a record series. An example is Finance/Accounts Payable/Invoices. For more information, refer to Chapter 3, "Retention schedules and file plans" on page 53.

Classification

Classification is a systematic identification and hierarchical arrangement of records into categories. It is the assignment of appropriate codes, categories, or any other descriptive information that can be used for retrieval, retention, or disposition.

Retention schedule (disposition schedule)

The *Retention schedule,* which is also known as the retention rules, specifies how long a record stays (is retained) in a phase and when the record transitions to the next phase.

A retention schedule can be driven by time, event, or event time. *Event time* means that when a specified amount of time has taken place after a particular event has happened, a record (in a records management system) has to move out of the current phase and into the next phase. The time does not start calculating until the particular event has taken place (for example, the closing of a case or project or the termination of a task or employee).

Generally, the retention is based on business and regulatory rules that require preserving records for a specific time period to comply with government laws and corporate policies.

Retention is determined by analyzing the following factors:

- Operational need
- Administrative need
- Fiscal need
- Legal need (statutory, regulatory, or contractual)
- Historical need

A records analyst gathers information about the records during a records inventory. This inventory provides information about the creation, use, and maintenance of the records, as well as potential retention requirements. Generally, laws and regulations provide a baseline retention requirement. If neither laws or regulations exist, it is up to the organization to determine the appropriate retention period after reviewing these factors.

Disposition

Disposition is the final phase of a record's lifecycle. When the retention period for a record ends, the record goes through the designated disposition process. Certain records are destroyed as part of the disposition process, while other records might be transferred to and thus permanently preserved in another archive system.

Hold

A *hold* is an action taken on records collections that indefinitely extend the records' retention period beyond what was originally assigned. A hold prevents the destruction of records. Records can be put on hold for any number of reasons, but generally, they are placed on hold due to audits, investigations, or litigation. When records are no longer being held, the records' original retention schedule and disposition schedule take control of the records.

Spoliation

Spoliation refers to deliberately withholding, deleting, or changing document content that is relevant to a legal case. You must avoid spoliation. Organizations must ensure that they do not alter the original format or context of the documents, and that they can also deliver the documents in an original or representative form.

Discovery

Discovery is part of litigation, during which each side is allowed to search for and review documents that are relevant to the current case. The discovery process consists of searching across all content (whether or not the content includes records) and identifying those records that match the discovery order usually when ordered by courts (and related to hold orders). After being filtered to remove any document that is under any attorney-client privilege or other classified restriction, and following negotiation with both sides of the legal issue on the terms and issues on which to search, the results must be made records (if they are not already), placed on a hold (so that they remain on hold until the end of the legal issue), and exported to opposing council (can be to a CD archive of a certain size in an agreed-upon format).

For more information about IBM solutions for eDiscovery, refer to:

`http://www.ibm.com/software/data/content-management/products/ediscovery`

2

IBM FileNet Records Manager system and architecture

In this chapter, we introduce IBM FileNet Records Manager and provide an overview of the product, including key business benefits, product highlights, and capabilities. We also discuss the architecture of IBM FileNet Records Manager as it relates to the IBM FileNet P8 Platform.

We cover the following topics in this chapter:

- Overview of IBM FileNet Records Manager
- System architecture
- Data model, workflow, and security
- User and administrative applications
- APIs and component integrator
- References

2.1 Overview of IBM FileNet Records Manager

IBM FileNet Records Manager is designed to uniquely combine content, process, and connectivity into a compliance infrastructure to automate and streamline records-based activities, eliminate unnecessary user participation, enforce compliance, and create business advantage through a compelling return on investment.

The key differentiator in IBM FileNet Records Manager is its innovative *ZeroClick* technology, which is designed to enforce records management policies without additional workload or participation from users. ZeroClick eliminates user-related error, reduces time and cost factors, and most importantly, ensures best practice records management.

IBM FileNet Records Manager supports the full *lifecycle* of records. Enforcement and compliance become both achievable and cost-effective.

IBM FileNet Records Manager also provides the flexibility to create single or multiple file plans for the purpose of managing records across the enterprise. A *file plan* represents a record classification and a storage schema that comprise a hierarchical structure of record management entities.

IBM FileNet Records Manager helps solve regulatory compliance challenges by:

- Automating the entire records management lifecycle process
- Invisibly enforcing consistent compliance and records management policy throughout an enterprise
- Organizing, securely storing, and quickly retrieving essential company records

As of the IBM FileNet Records Manager 4.5 release, this product is certified with the new Version 3 of the DoD 5015.2 Records Manager standard:

`http://jitc.fhu.disa.mil/recmgt/ibm/filenet/index.html`

IBM FileNet Records Manager 4.5 is certified for base capabilities and also for support of handling classified records. The previous versions of the product were certified with Version 2 of the DoD standard.

2.1.1 Key business benefits of IBM FileNet Records Manager

IBM FileNet Records Manager and its holistic approach to compliance deliver key business benefits across the enterprise:

- Reduced organizational risk
- Lowered operational costs (legal, administrative, and storage)
- Improved compliance with internal standards, as well as industry and governmental regulations
- Improved business processes
- Compliance as a competitive differentiator

Reduced organizational risk

IBM FileNet Records Manager and its innovative ZeroClick technology reduce organizational risk by automating and enforcing rule-based record capture, which minimizes or eliminates user involvement in the record-capture process. Automated capture eliminates one of the major sources of non-compliance.

IBM FileNet Records Manager also enforces consistent retention and disposition policies that are set by the organization's stakeholders. Involving the appropriate stakeholders ensures that record management is driven by legal, regulatory, and business considerations and that records are properly disposed when they are no longer needed for business purposes. Using IBM FileNet Records Manager also ensures that records are not accidentally destroyed, altered, or lost.

Lowered operational costs

IBM FileNet Records Manager helps reduce operational costs in a number of areas including legal, administrative, and storage. The consistent enforcement of retention and disposition policies ensures that information is retained only for the period that is required by legal, regulatory, or business requirements, and then the information is disposed. Enforcing retention and disposition policies not only directly impacts the storage costs that are associated with retaining the information, but more importantly, it reduces the risks and costs that are associated with legal discovery.

Furthermore, IBM FileNet Records Manager is built upon the IBM FileNet P8 integrated repository infrastructure, which greatly improves operational efficiencies and reduces costs associated with managing and synchronizing disparate repository silos. A single, unified repository allows you to quickly find and produce all relevant records.

Improved organizational compliance

Organizations live and operate in a regulated environment. These compliance requirements come both externally from laws and industry regulations, as well as internally from organizations' own business policies and practices.

IBM FileNet Records Manager offers a compliance infrastructure that allows organizations to approach their compliance needs from an enterprise perspective, as opposed to a series of point solutions that solve particular compliance requirements. By building records management capabilities directly into the infrastructure of the IBM FileNet P8 Platform, IBM FileNet Records Manager gives organizations the ability to evolve and adapt to changing compliance and business needs.

Improved and standardized processes

The IBM FileNet P8 Platform's tight integration between content, process, and compliance provides a powerful strategy for automating, enforcing, standardizing, and documenting business processes across the enterprise, which is a key requirement for any compliance strategy.

IBM FileNet Records Manager's process-centric approach to compliance allows an organization to not only easily adapt the processes to a shifting compliance environment, but it also provides the revision history of the processes – another key compliance requirement.

Compliance as a competitive differentiator

A properly implemented compliance solution can be turned from a corporate obligation to a corporate strategy. Automating, enforcing, and standardizing your business process gives you the opportunity to model and optimize these processes. The less time spent on manual compliance, the lower the costs, the lower the risk of non-compliance due to human error, and the less time spent with regulators and auditors.

2.1.2 Product highlights and capabilities

IBM FileNet Records Manager is a fully functional records management system that supports all of the primary records management functionality, including:

- File plan design and creation
- Record declaration and classification
- Record retention and disposition
- Management of electronic and physical records
- Record search and retrieval
- Record holds
- Auditing and reporting

IBM FileNet Records Manager is much more than just a records management system. Because of its integration with IBM FileNet P8 Platform, the benefits and capabilities are much greater than the sum of its parts, and they include:

- Compliance infrastructure (not a point solution)
- Active compliance
- Fully integrated architecture and repository
- ZeroClick automated records capture
- Federated records management and search

IBM FileNet Records Manager combines content, process, and compliance services into an infrastructure that is built upon a single, fully integrated repository that manages all of your content (documents, e-mails, and records). IBM FileNet Records Manager provides the ability to perform consolidated searches and retrievals. Compliance policies are uniformly enforced at the technology layer and do not have to rely on action by business users or records managers. Using the same event-based architecture of the IBM FileNet P8 Platform that allows active content in a business process, IBM FileNet Records Manager creates an active compliance infrastructure using ZeroClick technology.

2.1.3 ZeroClick

ZeroClick ensures best practice records declaration, classification, and records administration while minimizing the impact on the business users and records administrators. ZeroClick streamlines the declaration and classification of records based on user events and actions, such as the filing of an electronic document into a folder, or internal and external system events, such as the receipt of a transaction from another system. ZeroClick is designed to eliminate user interactions, thus minimizing user-related errors, time, and cost factors. With ZeroClick, records are active elements as opposed to passive elements that are tightly integrated with your business applications.

2.1.4 Capturing and declaring records

In an enterprise, content can come from a variety of sources, including scanned images, faxes, e-mails, electronic documents, eForms, and Web content. It is imperative that any enterprise content management (ECM) platform provides mechanisms for easily ingesting content from disparate content sources and transparently manage them as records in a file plan. IBM FileNet P8 provides a broad set of capabilities and integrated product suites for ingesting content from a variety of sources and automatically declaring and classifying them as records in the file plan. These products typically target a particular content source:

- IBM FileNet Capture: Works with scanned images and faxes

- IBM FileNet Email Manager: Works with e-mails from e-mail servers
- IBM FileNet Records Crawler: Works with electronic documents on network file shares
- IBM Content Collector: Works with both e-mails from e-mail servers and electronic documents on network file shares

> **Important:** During the writing of this book, both IBM FileNet Email Manager and IBM FileNet Records Crawler have been replaced by IBM Content Collector.

- IBM FileNet Business Process Manager (BPM): Works with documents attached as part of business processes
- IBM FileNet Workplace and IBM FileNet Workplace XT: Works with any electronic document created or managed by a user
- IBM FileNet Application Integration component: Works with documents authored and captured directly from the Microsoft Office product suite
- Workplace Entry Templates: Allow an administrator to customize the declaration wizards providing default values, automating data collection, and minimizing user input

Figure 2-1 shows how these products provide sources for capturing business records in IBM FileNet Records Manager.

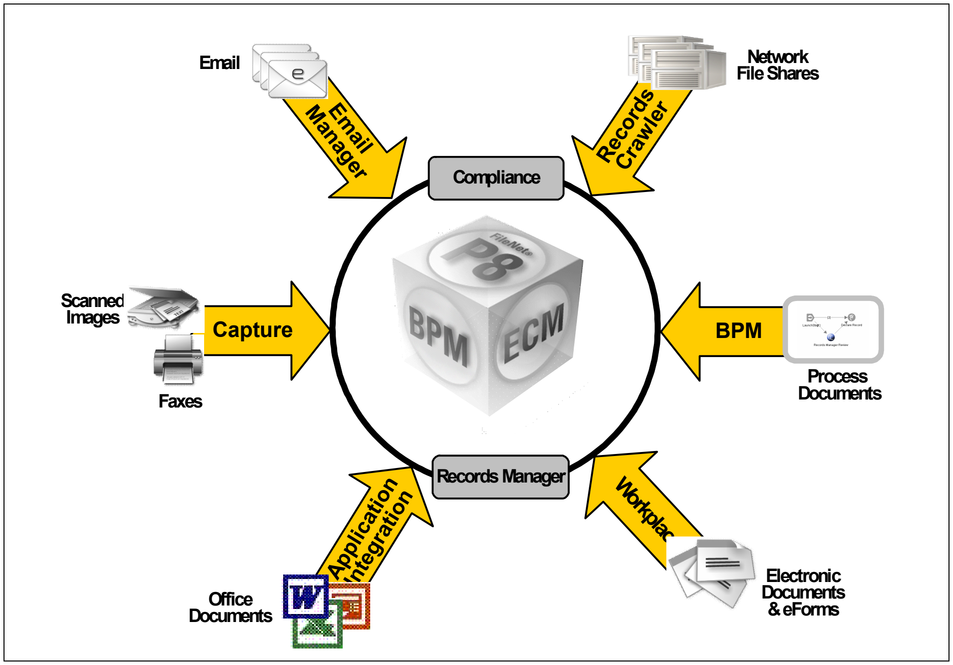


Figure 2-1 Records are captured into IBM FileNet P8 from different sources

2.2 System architecture

In this section, we discuss the system architecture of IBM FileNet Records Manager and review the major components that make up the IBM FileNet Records Manager system.

2.2.1 IBM FileNet Records Manager as an extension of the IBM FileNet P8 Platform

The key to understanding IBM FileNet Records Manager system architecture is to first understand the underlying IBM FileNet P8 architecture.

At the core of the IBM FileNet P8 architecture, there are three engines:

- *Content Engine* (CE): Manages business-related content by providing a repository for documents and providing constructs, such as folders and custom objects, to help organize and manage the content. Content Engine provides library services, lifecycle management, search capability, archive, and security services. It handles database transactions required for managing one or more object stores. An *object store* is a repository for storing objects in an IBM FileNet P8 environment.

- *Process Engine* (PE): Enables users to create, modify, and manage business processes. Process Engine provides process services, supports business rules for processes, and provides e-mail notification capability. In addition, there are the Process Analyzer (PA), Process Simulator (PS), and Orchestration applications for monitoring and tuning the business processes.
- *Application Engine* (AE): Hosts the Workplace or Workplace XT Web application, hosts Java applets, and supports Content Engine operations using content and process application programming interfaces (APIs). It is the presentation tier for both process and content.

IBM FileNet P8 Platform leverages a multi-tier distributed architecture that combines these capabilities, along with flexible enterprise connectivity, to facilitate highly scalable and high performance solutions. Based on an open Java 2 Platform, Enterprise Edition (J2EE™) environment, IBM FileNet P8 Platform uses XML, SOAP, and other standards to integrate with enterprise business applications and content. IBM FileNet P8 Platform includes comprehensive APIs, including SOAP-based Web services for building solutions using a service-oriented architecture (SOA).

Figure 2-2 illustrates the high-level system architecture of IBM FileNet P8 Platform. It shows the relationship between the three core engines and where the engines reside within a typical *n*-tier distributed architecture.

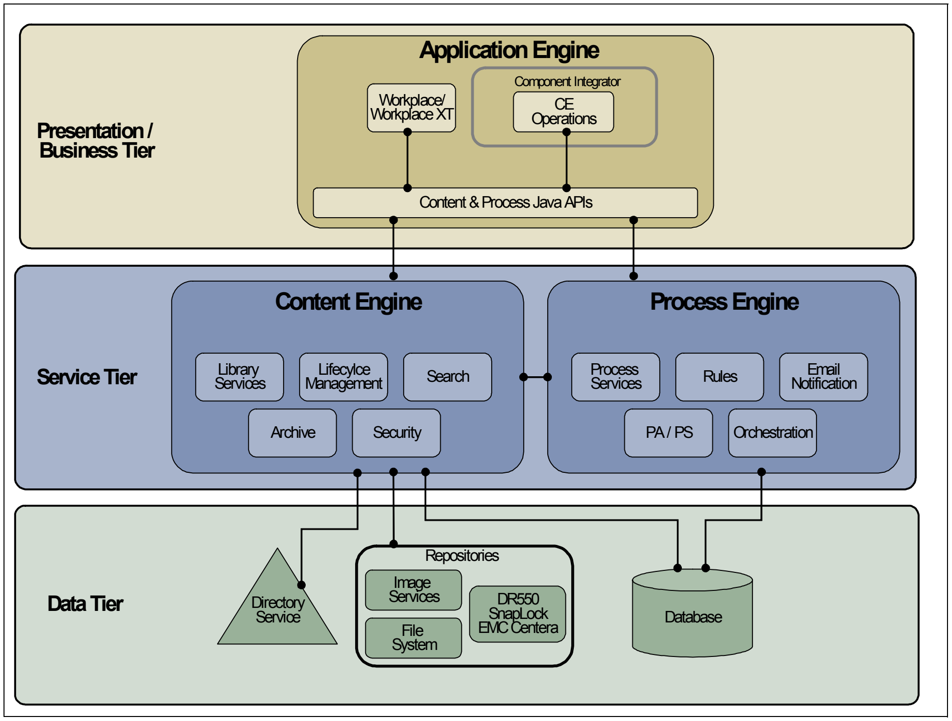


Figure 2-2 High-level architecture of the IBM FileNet P8 Platform

IBM FileNet Records Manager builds on top of IBM FileNet P8 Platform; it leverages and extends the services provided by the core engines of IBM FileNet P8 Platform. IBM FileNet Records Manager provides records management functionality, with a single repository that stores all electronic documents and records. With IBM FileNet Records Manager, you can automate the management of electronic and physical records at the enterprise level.

Figure 2-3 shows several of the major IBM FileNet Records Manager components within the IBM FileNet P8 architecture and their relationship to the underlying core IBM FileNet P8 Platform services.

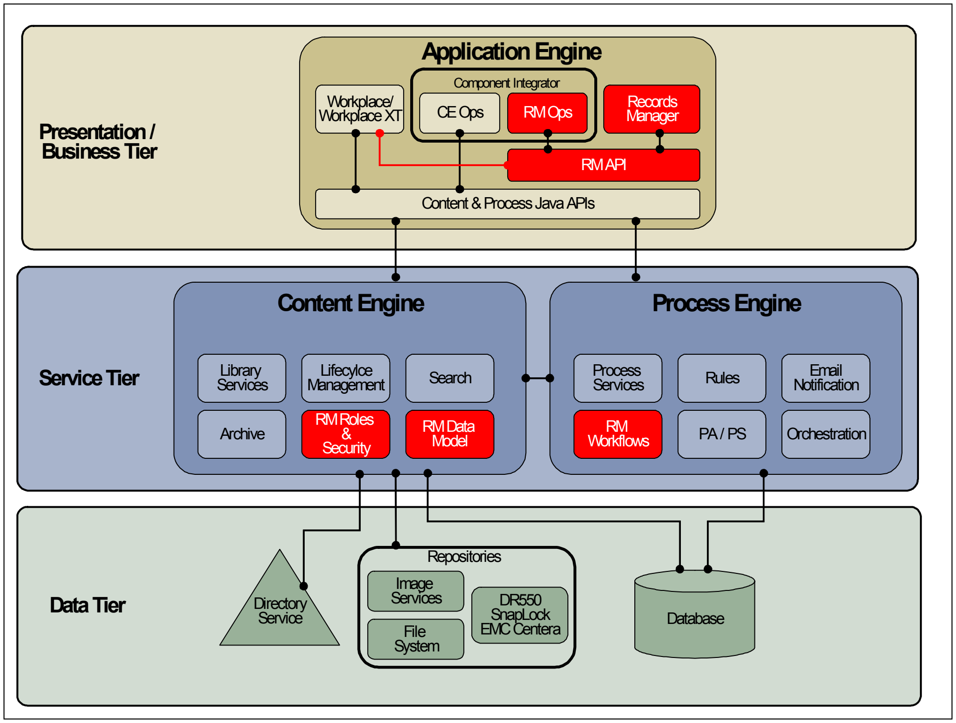


Figure 2-3 IBM FileNet Records Manager architecture as an extension of the IBM FileNet P8 Platform

As shown in Figure 2-3, several of the major IBM FileNet Records Manager components are:

- Within Content Engine:
  - IBM FileNet Records Manager data model: This component provides the core definitions for IBM FileNet Records Manager business objects, such as record classes, records folders, and disposition schedules upon which the records management system is built.
  - IBM FileNet Records Manager roles and CE security: The records management security capabilities are built upon the underlying Content Engine security model coupled with default IBM FileNet Records Manager security roles that determine functional user access. IBM FileNet Records Manager takes advantage of CE marking sets to implement certain features of its security model.
- Within Process Engine:
  - IBM FileNet Records Manager workflows: PE provides special records management-related workflows for implementing a variety of disposition

actions. Workflows leverage the full capability of PE and can be completely customized.

- Within Application Engine:
  - IBM FileNet Records Manager Web application: This component provides core administrative and management functions for the file plan and the records that it contains.
  - IBM FileNet Workplace Web application: Workplace provides a comprehensive user interface and user access to documents and other business objects, as well as access to any associated business processes. Workplace is integrated with IBM FileNet Records Manager for automated and manual record declaration.
  - IBM FileNet Records Manager Java API: This API exposes the IBM FileNet Records Manager functions for custom application development.
  - Component integrator (IBM FileNet Records Manager operations): This component integrates records management functionality into a BPM environment and thus records-enables business processes.

We discuss these components in more depth in the later sections of the chapter.

2.2.2 Records stored as Content Engine objects

The IBM FileNet P8 family of products uses a common object model (or data model) managed by Content Engine that leverages *object-oriented* design to store and manage content. IBM FileNet Records Manager is built directly into Content Engine and therefore inherits the underlying object model and object-oriented design of Content Engine. Information stored and managed in the system are represented as objects, described through the object properties (metadata), identified by object classes, and associated with operational methods of the objects. These objects reside in IBM FileNet P8 content repositories, also known as *object stores*. The object stores are managed by Content Engine.

For IBM FileNet Records Manager configuration, there are two types of object stores:

- Records-enabled content Object Store

  The Records-enabled content Object Store (ROS) serves as the content repository for electronic documents. Documents stored in an ROS can be declared as records.

- File Plan Object Store

  The File Plan Object Store (FPOS) serves as the object store for the file plan, records categories, disposition schedules, and all other business objects required to manage records. When documents in the ROS are declared as records, the record-related information (metadata) is stored as a separate record object in the FPOS.

Figure 2-4 illustrates IBM FileNet Records Manager integrated with the IBM FileNet P8 Platform, the relationship between records (which store the record-related metadata of the declared documents) in the File Plan Object Store (FPOS) and the associated declared documents in the Records-enabled content Object Store (ROS).

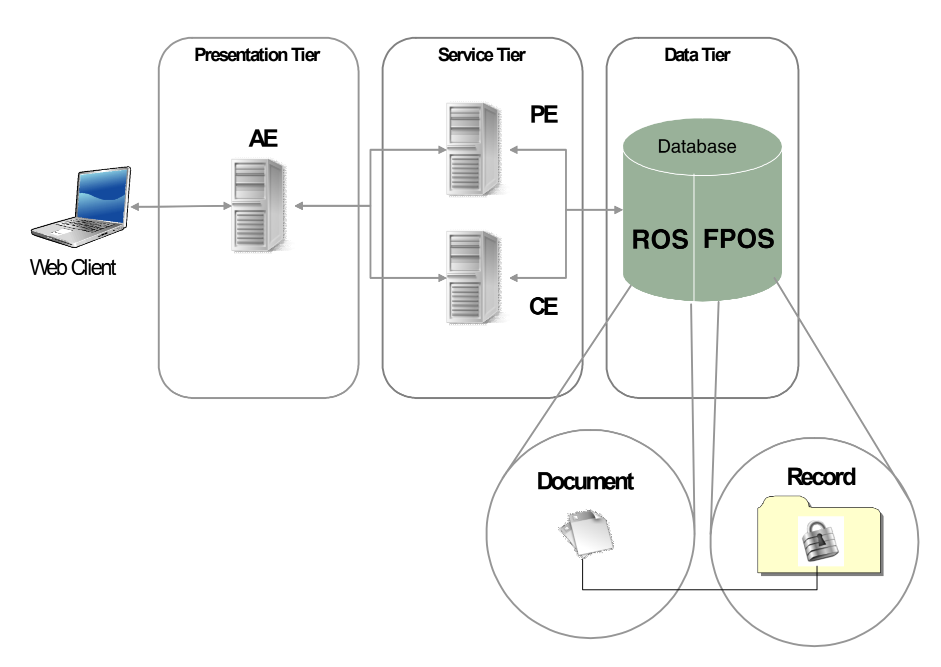


Figure 2-4 Integrated file plan and content repository architecture

As illustrated in Figure 2-4, an IBM FileNet Records Manager record is a completely separate object from the associated declared document. The record must be contained in a file plan within FPOS. The record object has a direct reference to the actual declared document that exists in the ROS (or a reference to a physical artifact that exists outside of the ROS). The record object acts as a security proxy for the declared document that it references. For documents that reside in an IBM FileNet P8 content repository, the repository is simply another object store that is records-enabled and is managed by Content Engine. Other electronic documents can reside in a repository outside of IBM FileNet P8 content repository. In the case of physical records, the records in IBM FileNet

Records Manager serve as markers or references to the physical objects with relevant information to identify and manage the physical objects.

This architecture allows IBM FileNet Records Manager to easily unify and manage records across disparate, heterogeneous repositories, including:

- Native IBM FileNet P8 electronic documents
- Non-native electronic documents (external repositories)
- Physical records or artifacts

Figure 2-5 shows that IBM FileNet Records Manager manages records from various repositories.

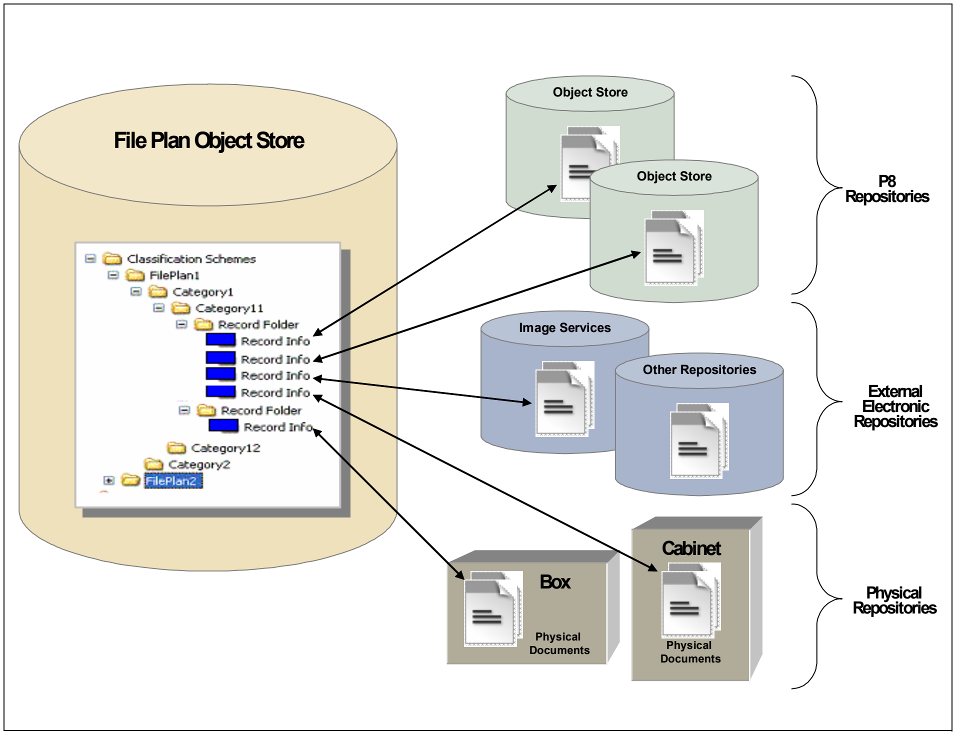


Figure 2-5 IBM FileNet Records Manager manages records across disparate repositories

2.3 Data model, workflow, and security

In this section, we introduce data model options in IBM FileNet Records Manager, the available workflows, and briefly discuss IBM FileNet Records Manager security roles.

2.3.1 IBM FileNet Records Manager data model

Content Engine provides the object-oriented content repository. Everything in the repository is represented as objects with associated metadata. Documents, folders, custom objects, and other business objects are all objects within Content Engine. IBM FileNet Records Manager entities are business objects defined in and managed by Content Engine.

A key component in IBM FileNet Records Manager is the data model. Using the object-oriented design of Content Engine and extending its object model, the IBM FileNet Records Manager data model provides the core definitions that support the implementation of IBM FileNet Records Manager business objects. The data model is an abstraction layer that acts as a template to provide the initial imprint or starting point for an IBM FileNet Records Manager system.

Data model options

IBM FileNet Records Manager supports four data model options:

- Base
- DoD
- DoD Classified
- PRO

DoD refers to the Department of Defense standard 5015.2 for records management software. Each chapter in DoD addresses separate levels of the standard. PRO stands for the UK Public Record Office's similar standard, which is now called The National Archives 2 standard or TNA2.

The *Base data model* is the core IBM FileNet Records Manager data model option that implements all required Records Management functionality. The Base data model is the data model that is used by most customers unless they require the specialized functionality of the other data models. Currently, for federation, IBM FileNet Records Manager works with Content Federation Services using the Base data model only.

The *DoD data model* is a special variation of the Base data model option. It includes configurations specifically intended for DoD product certification. Only consider this data model option if specific DoD-related configuration is required.

The *DoD Classified data model* implements the functionality specified in classified records chapter of the DoD standard (Chapter 4 in Version 2 and Chapter 5 in Version 3). DoD Classified is a superset of Chapter 2 and addresses requirements and functionality for the management of classified records. Only consider this data model option if customers require Security Classification support.

The *PRO data model* implements the functionality and behavior specified in the PRO standard.

Best Practice: Customers, who want to implement standard records management functionality and behavior, need to install the *Base data model*. We do not recommend using other data model options unless you have specific requirements that the Base data model does not address.

Note: At the time of writing this book, only the Base data model supports localization.

Business objects in FPOS

The IBM FileNet Records Manager data model encapsulates the underlying records management business objects and associated metadata. The business objects and their associated metadata are defined within the File Plan Object Store (FPOS).

High-level records management business objects include:

- File plan
- Record containers:
  - Record categories
  - Records folders
  - Record volumes
- Record objects
- Disposition schedules
- Disposition events
- Searches
- Holds
- Security classification guides

These objects are the building blocks for a records management system. You can also create custom object definitions that are relevant to your business context and requirements.

2.3.2 IBM FileNet Records Manager workflows

IBM FileNet Records Manager is built upon IBM FileNet P8 Platform and leverages the capabilities of Process Engine. Like many areas of a business, records management is based on process; IBM FileNet Records Manager, therefore, is strongly process-centric.

To perform records management tasks, such as disposing of records, reviewing records before they are destroyed, and destroying records, IBM FileNet Records Manager provides the following workflows:

- Disposition Review workflow
- Cutoff workflow
- Create Record Folder workflow
- Destroy workflow
- Export workflow
- Interim Transfer workflow
- Physical Record Management workflow
- Screening workflow
- Transfer workflow
- Vital Record Review workflow

Note, while each of these workflows can be used as is, each preconfigured workflow is usually customized to reflect your own internal business processes.

2.3.3 IBM FileNet Records Manager roles and Content Engine security

One of the primary functions of a records management system is controlling access to the records and information managed by the system. When a document is declared as a record, the document is under the control of the records management system. The access rights specified in the file plan override any access rights that were specified in the document before it was declared as a record.

IBM FileNet Records Manager security settings determine the groups and users who can access record entities, including file plans, categories, folders, volumes, and records. Security settings also control the permissions that are granted to each group or user.

Appropriate security controls for records ensure that:

- Only authorized users can access the appropriate records.
- Records are destroyed as a result of a defined records disposition policy. Authorized records personnel can delete records only under rare circumstances, such as wrongly classifying non-records materials to be records. Records users cannot delete records accidentally.
- Users can perform only those operations on records for which they have access rights granted.

IBM FileNet Records Manager leverages the core security model that is used with the IBM FileNet P8 Platform. The security model supports existing security standards, such as Lightweight Directory Access Protocol (LDAP) and Secure Sockets Layer (SSL), as well as commonly used directory services products. IBM FileNet Records Manager provides an LDAP directory-based authentication model and a comprehensive auditing system.

IBM FileNet Records Manager uses a number of roles to initially configure the security for the various records management entities. These roles serve to define the functional access rights for users and groups. The standard roles defined by the Base data model are:

- Records Administrator
- Records Manager
- Records Privileged User
- Records User

These roles are typically tied to groups in the directory service.

The predefined roles vary with other data models. For example, the PRO data model implements a Records Reviewer role in place of the Records Privileged User role. The DoD Classified data model implements an additional role called the Classification Guide Administrator.

For a more detailed discussion of these roles and security, refer to Chapter 4, “Security” on page 85.

2.4 User and administrative applications

IBM FileNet Records Manager provides Workplace and the IBM FileNet Records Manager Web application as user and administrative Web applications. In addition, IBM FileNet Records Manager provides stand-alone applications, such as Disposition Sweep, Hold Sweep, and File Plan Import Export Tool for administrative operations.

2.4.1 Workplace

Workplace is a Web application that provides users an interface into the IBM FileNet P8 system. It is both content-enabled and process-enabled. It allows users to access and manage their documents, as well as perform business process tasks, through the application’s inbox portal.

Workplace is the primary mechanism by which day-to-day records users interact with IBM FileNet Records Manager (if no custom applications have been created). However, the degree to which users are aware of IBM FileNet Records Manager is in part dependent upon whether you want to expose IBM FileNet Records Manager to users. Best practices typically dictate minimizing or completely masking the users' knowledge of the underlying IBM FileNet Records Manager system. You can set up IBM FileNet Records Manager to be completely transparent to users through ZeroClick technology.

Typical user scenarios involve searching for documents, creating new documents, versioning existing documents, or performing business process tasks. Whether these documents are declared as records can be transparent to users as they carry out these tasks.

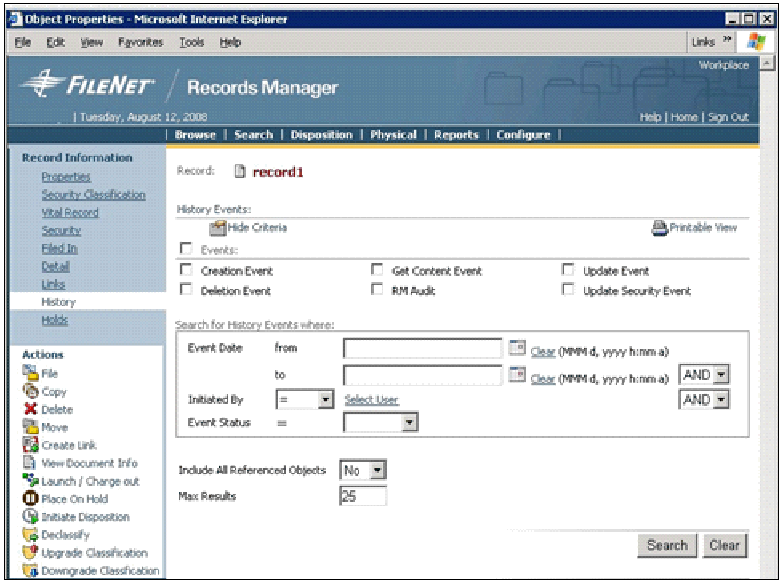
You can control how many IBM FileNet Records Manager functions you want to expose to users. It might be desirable in certain scenarios to give users control over the record declaration process. You can configure Workplace to give users various levels of control that range from:

- Full control, where users can decide when and where to declare documents as records
- Intermediate control, where users can decide when to declare documents as records, but IBM FileNet Records Manager controls where to declare them as records
- No control, where the record declaration process is completely automated and thus is transparent to users

2.4.2 IBM FileNet Records Manager Web application

IBM FileNet Records Manager Web application is the key user interface that is used to configure and maintain the file plan and to manage the records in the file plan. This application is typically used by Records Managers and Records Administrators, although it can also be used by other users for search and retrieval and to perform a variety of other manual operations.

Figure 2-6 shows the user interface of the IBM FileNet Records Manager Web application.



Record: record1

□ Events:

□ Creation Event □ Get Content Event □ Update Event
□ Deletion Event □ RM Audit □ Update Security Event

Event Date from

to

AND

Initiated By =

AND

Event Status =

Include All Referenced Objects No

Max Results 25

Figure 2-6 IBM FileNet Records Manager Web application user interface

The IBM FileNet Records Manager Web application provides the following administrative and management functions:

- Configure and maintain the file plan.
- Search for records and file plan entities.
- Define and set disposition schedules and retention triggers.
- Define and apply record holds.
- Apply security settings to file plan entities.
- Initiate disposition for records or record containers.
- Provide auditing and reports.

We discuss many of these features and step-by-step instructions in the subsequent chapters in this book.

2.4.3 Disposition Sweep

Disposition Sweep is a stand-alone application that calculates the disposition phase information for records and other entities in a file plan. It primarily updates information about record entities. It is also responsible for automatically initiating several of the optional workflows, such as the Screening workflow and the Cutoff workflow.

Disposition Sweep is often run as a background process and can be run periodically by the scheduling services of the operating system.

2.4.4 Hold Sweep

Hold Sweep is a stand-alone application that finds records meeting the conditions specified in one or more conditional holds and automatically places these records on hold. A conditional hold allows you to specify metadata-based search criteria upon which to evaluate whether a record needs to be placed on hold.

Hold Sweep typically is used at the onset of a legal action or a regular audit process where there might be hundreds or thousands of documents scattered throughout the file plan that need to be placed on hold. A conditional hold is created with the appropriate search criteria, and Hold Sweep is subsequently run against the conditional hold. As new documents are ingested and declared as records, several of the new documents might also require to be placed on hold based on the search criteria. Each time that Hold Sweep is run, it automatically places on hold all of the records that meet the specified hold criteria, thereby freeing the records management staff from having to periodically search for these documents and manually place them on hold.

Like Disposition Sweep, Hold Sweep is often run as a background process and is typically run periodically by the scheduling services of the operating system.

2.4.5 File Plan Import Export Tool

The File Plan Import Export Tool is a stand-alone application that allows an administrator to move a File Plan and its associated objects (such as schedules, holds, events, and locations) to another object store. You can use the File Plan Import Export Tool to move a production file plan to a test environment or to another server. It can also be useful in deploying a file plan to multiple FPOS if required.

2.5 APIs and component integrator

In addition to the readily available user and administrative applications, IBM FileNet Records Manager offers APIs and a component integrator that you can use to create or customize your IBM FileNet Records Manager applications.

2.5.1 IBM FileNet Records Manager and Bulk Declaration Services

There are two API sets available for custom application development related to records manager functions.

IBM FileNet Records Manager API

IBM FileNet Records Manager provides an application programming interface (API) that exposes all of IBM FileNet Records Manager functionality for custom application development. The API is Java-based and is built upon the IBM FileNet P8 Platform's Content Engine Java API.

Bulk Declaration Service (BDS) API

The BDS API is available for the high performance, large volume ingestion of records into IBM FileNet Records Manager. A primary use case for BDS is a large-scale migration and the conversion of records and content from existing records and content management systems into the IBM FileNet Content Manager repository.

BDS provides the following functions:

- Bulk declaration of new physical records
- Bulk declaration of new electronic records for existing documents in the IBM FileNet P8 content repository
- Bulk creation of new documents in the IBM FileNet P8 content repository and optionally declaring them as records

2.5.2 IBM FileNet Records Manager component integrator

IBM FileNet Records Manager uses a process-centric compliance and records management approach. The IBM FileNet Records Manager component integrator provides a mechanism for record-enabling business processes. Figure 2-7 shows an example claims business process that includes record declaration of associated claim documents as an integral part of the process. As a final step in this example, records are declared and automatically classified into the file plan.

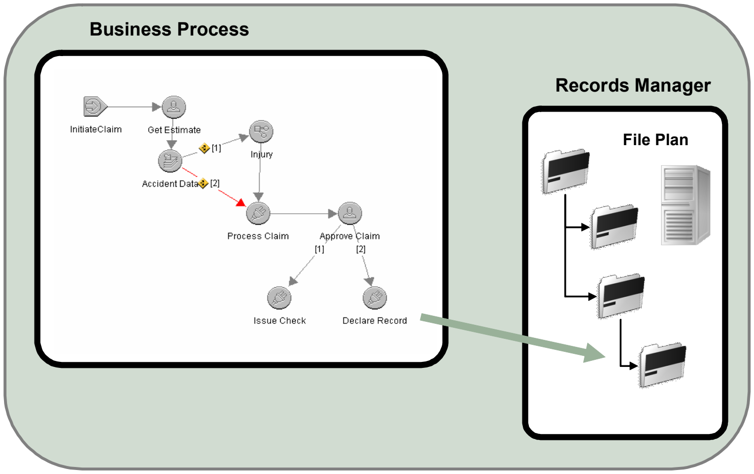


Figure 2-7 Records-enable business process using BPM and component integrator

The IBM FileNet Records Manager component integrator integrates directly with Business Process Manager (BPM) and provides records management capabilities within business processes, including:

- Record declaration
- Record folder creation
- Record destruction
- Transferring or exporting records

You can further extend the integration of IBM FileNet Records Manager and BPM by building custom Java components using the IBM FileNet Records Manager API to expose more specialized and tailored records management capabilities directly into business processes.

> **Best Practice:** Fully integrate your records management capabilities with BPM to ensure a process-centric approach to compliance.

2.6 References

Visit often and become familiar with the many resources and sources available to you to provide more examples, samples, and updated options, including:

- Main product page:

  http://www.ibm.com/software/data/content-management/filenet-records-manager/

- Product support page:

  http://www.ibm.com/software/data/content-management/filenet-content-manager/support.html?S_CMP=rnav

- Product documentation page:

  http://www.ibm.com/support/docview.wss?rs=3278&context=SSNVNV&context=SSTHRT&context=SSNVUD&context=SSS236&context=SSNW2F&uid=swg27010422&loc=en_US&cs=UTF-8&lang=en

- Product forum and user group
- Various dynamic online records communities and social networking groups

3

Retention schedules and file plans

In this chapter, we introduce the topics of retention schedules and file plans. We also introduce key records management terminology and map these terms to IBM FileNet Records Manager concepts.

We discuss the following topics:

- Retention schedules
- Retention schedule planning and creation
- File plans
- Creating and maintaining a file plan
- File plan performance considerations

3.1 Retention schedules

A *retention schedule* is a timetable that specifies the length of time that records must be retained before disposition and the disposition action that must be performed on the record. A retention schedule describes a company's records, their ownership, and regulatory citations, as well as the disposition period based on legal, regulatory, and business needs. It is a *living* document within an organization that must be reviewed regularly.

> **Note:** *Retention schedule* can be a confusing term. Sometimes, records managers use this term differently from how it is used within records management software. In this chapter, the term is defined from the perspective of how a records manager uses it, namely to mean a document describing how all of a company's regulated records are managed. Table 3-2 on page 60, Sample Retention Schedule, demonstrates a simple example of a retention schedule. Each line within this sample equates to what records managers typically call a *retention rule*. A typical retention schedule can have hundreds of retention rules.
>
> The confusion occurs, because records management software uses the term synonymously with retention rule and disposition schedule. Specifically, in IBM FileNet Records Manager, a single disposition schedule object typically represents a single retention rule, not the entire retention schedule.

Most companies have a form of retention schedule or a set of retention rules for their records. If your company has a retention schedule, review and revise it according to the various requirements that we describe here. If you do not have a schedule yet, you need to create one.

The following requirements can drive a retention schedule:

- *Compliance* and *regulatory* requirements. Industries and government regulations often impose special retention requirements for records.

  > **Note:** *Compliance* is the act of adhering to and demonstrating adherence to internal or external regulations. Either a federal or state agency issues a *regulation* as directed by legislation. A regulation is a compromise between prohibition and no control at all.

- *Fiscal* requirements on record keeping.
- *Business* requirements, which can include audit, company's retention policy, legal counsel opinion, or business continuity reasons.

- Administrative need for the record.
- Historical need for the record.

To determine the retention period of a record, stakeholders from legal counsel, compliance officers, and business users need to be involved, because the requirements vary among countries, states, municipalities, industries, compliance and legal jurisdictions, and document types. The retention schedule is updated continually as business and regulatory needs evolve.

> **Important:** Companies are responsible for ensuring their own compliance with relevant laws and regulations. It is the company's responsibility to obtain advice from competent legal counsel concerning the identification and interpretation of any relevant laws that can affect the company's business and any actions that the company might need to take to comply with these laws.
>
> IBM does *not* provide legal, accounting, or audit advice or represent or warrant that its services or products will ensure that a customer is in compliance with any law.

There are usually multiple retention rules associated with a retention schedule. Each *retention rule* specifies how long records are retained and what to do after the retention period expires. Each retention rule applies to a specific group of records.

3.2 Retention schedule planning and creation

Creating a retention schedule for a company is one of the most critical tasks in records management. To plan and create a retention schedule for a company, you can use the following exercises and activities as guidelines:

- Understanding company's records management policy
- Understanding a company's records management procedures
- Understanding regulatory requirements
- Conducting records inventory
- Creating the retention schedule

3.2.1 Understanding company's records management policy

A *records management policy* is based on legal, regulatory, and business requirements. If the company does not have a records management policy, it is a best practice to develop one. A policy is a living document and must be reviewed and revised as your business needs evolve. In addition, everyone in the company needs to adhere to the policy. Records management procedures are developed in accordance with the policy. The company can use technology to enforce records management policies by attaching records management policies to the documents that reside in the repositories.

Example 3-1 provides an example of a part of a records management policy.

Example 3-1 Records management policy example

Essential information is information deemed necessary for satisfying corporate legal requirements for conducting business operations.

Essential information is to be retained long enough and in appropriate media to meet the laws of the countries in which this corporation conducts its operations, retrievable in a usable form throughout the retention period and disposed when the retention period expires, unless subject to a hold order.

3.2.2 Understanding a company's records management procedures

Records management procedures are developed in accordance with the records management policy. These procedures provide the operational task, role, and outcome that are required to ensure that the company adheres to the policy and the associated retention schedule.

If a company does not have a set of procedures for records management, it is a best practice to develop one. Procedures are specific to the industry to which the company belongs, the nature of the business, and the operation of each area within the company.

Example 3-2 provides an example of a part of a records retrieval procedure that might apply to the management of physical records.

Example 3-2 Records retrieval procedure example

All retrievals must be returned in 60 days or less. If needed for a longer period, the Records Administrator is to be notified. If permanent removal is requested, management documentation to confirm retention will be required. See Process Contacts.

- Access the Records Retrieval Form at web address: ...
- Complete all of the required fields:
  - Enter the box number you are requesting. If a file is being requested, complete the "File Number" field under this section as well.

- Be sure to review the Priority selection, because it will default to "Normal" service. If urgent, same-day delivery is required, select the emergency option.

- When the form is completed, follow these instructions: 1) Save the document as a Word file. 2) After the file is saved, prepare an e-mail to the Administration Team with the supply form included as an attachment.
- Administration Team will review the contents of the form.
- You will receive a response in 48 hours (via e-mail) to advise that your order has been processed. Administration Team will provide an order number and expected day/time for delivery.

Note: Many existing retention schedules and records management procedures are written for managing physical records. Many times, an organization must revise and clarify these procedures and schedules when trying to implement an electronic records management system.

3.2.3 Understanding regulatory requirements

Many countries have legislation regarding records keeping. Most laws are applicable to physical and electronic records. Certain laws specify the active and inactive retention period, and certain laws have special compliance requirements about storage. Regulatory requirements vary from country to country, industry to industry, and for various legal entities, states, and document types.

3.2.4 Conducting records inventory

Companies need to conduct an *inventory* of their documents. The inventory can be at the departmental level and can be conducted through questionnaires. A document inventory is a list of documents that exist within a company. The inventory provides information that describes the characteristics of records that are created or captured.

Table 3-1 on page 58 shows an example of a records inventory template. The list that we show here is not meant to be exhaustive.

Table 3-1 Sample record inventory template

Record code:	Record owner:
Record description:	Department:
Format:	Retention period:
Location:	Disposition:

After performing a document inventory, you can categorize them into the appropriate categories. Users must include all types of documents, including paper records, microfiche, electronic documents, electronic mail, fax, instant messaging, collaboration content, voice recording, wireless communication content, audio, video, shared drive content, and Web content.

Reviewing the existing records filing process

Many companies already have a records filing process. Review and study the existing records filing process. The goal of this review is to accomplish the following tasks:

- Identify records that are being generated within the company.
- Understand the hierarchical order among groups of records.
- Ensure that all records from the company are presented and collected.
- Identify who created the records and how they are created.

This review gives you a better understanding of what records the company has and helps you to create the retention schedule and the file plan.

3.2.5 Creating the retention schedule

After gathering a complete records inventory, records are categorized into groups of related records that can be filed as a unit for retention purposes. The retention period for each group needs to be based on legal mandates, regulatory requirements, business requirements, and good business practices. For example, SEC 17a-4 requires that specific records retained by brokers and dealers must be preserved for at least six years, the first two years in an easily accessible place, while other records must be retained for at least three years, the first two years in an easily accessible place. Documents falling into the first requirement then must be kept for at least six years. Based on regulatory requirements and other business requirements, you can categorize that special group of records into one unit. Make sure that the records are logically grouped together (from the company's business operations perspective) *and* that they all have the *same* retention period based on requirements.

In addition to determining the proper retention period for records, you also have to determine the action by which to dispose of records when their retention period expires. This determination is important, because proper disposition of records protects companies from future liability and helps to control storage requirements. Records disposition actions include (but are not limited to) destroying the records and archiving records to another record's holding authority. You can also set up the system to review records when their retention period expires and then decide what to do. By design, a records management system does not typically destroy records automatically when records reach their retention period, even when the disposition is set to destroy. A records management system can enforce human verification before anything is done to the expired records.

Important: Retention period and records disposition actions are two of the key elements in a retention schedule. Proper understanding of your company's records policy, records procedures, and the proper understanding and *correct interpretation* of laws and regulations are crucial in finally determining the length of time that a particular group of records must be kept and what to do with the records afterward. You must involve legal counsel and special records professionals in developing the actual retention schedule for a company.

For more information about records disposition, refer to Chapter 6, "Records disposition" on page 149.

Table 3-2 is an example of items in a retention schedule.

Table 3-2 Sample retention schedule

Record series	**Description**	**Citation**	**Total retention period**	**Disposition**
Claim files	Insurance claim documents. (Example: claim forms, e-mail, voice recordings, regular mail, photos, and video.)	10 CCR 260.241.1	Claim close date + 5 years	Destroy
Employee records	Employee related documents including employee information, employee related activity, and activity of associated persons. (Example: Employee files, finger printing, compensation, salary, benefits, registration, and licensing.)	16 CCR 607	Termination date + 7 years	Destroy

Here is a list of suggested fields in a retention schedule (the list is not meant to be exhaustive):

- Record series: A records management term that means a group of related records that can be filed as a unit for retention purposes. It can be further broken down into primary, secondary, and tertiary record series if required.
- Series title: The name by which the group of records is known by the users.
- Description: Defines the scope of the records included in the category.
- Office of record: Refers to the owner, which can be a department responsible for maintaining the official records for the retention period.
- Vital record: An identifier to indicate whether the record is needed in the event of disaster. Vital records are usually stored off-site and replicated for disaster recovery purposes.
- Active retention period: The retention period for records required for current use.
- Inactive retention period: The period of time during which inactive records must be maintained by the company.

> **Note:** *Active* records are consulted routinely in the daily performance of work. *Inactive records* are rarely used, but they must be retained for occasional reference or to meet audit or legal obligations. An inactive record is a concept used mainly with physical records: inactive indicating that the record is eligible to be moved to a storage warehouse. Distinguishing between active and inactive records is less relevant when dealing with the management of electronic records.

- Total retention period: The sum of the active retention period and the inactive retention period.

> **Note:** The retention period includes information about the basis on which the time period can be calculated. For example, you might have a retention period of 10 years, but for certain records, this period can be 10 years from the storage date, and for other records, it might be 10 years from the date superceded.

- Citation: The statutory authority or law that governs the retention of the records. For example, SEC 17cfr275.204-2(e)(3) requires that customer communications are retained for five years.
- Disposition: The final action for a records series. Examples of valid actions are *destroy* (physically destroying the records) and *accession* (or archiving, which is transferring records to other record's holding authorities).
- Medium (media type): The object or device on which the record resides. Examples of medium are paper, microfilm, computer disk, and CD-ROM.
- Records series code: Can be used to reference a citation schedule or a disposition schedule, or to uniquely identify the series with a standard reference code.

You use the retention schedule as one of the major inputs in the design of the file plan and disposition schedules that need to be implemented within an electronic records management system (ERMS).

3.3 File plans

A *file plan* is a structured filing schema, which is normally based on a business classification scheme, that a records management system uses to support a retention schedule. There is no universal file plan for all companies. Each file plan is unique and depends upon the types of businesses with which the company or organization deals. A file plan specifies how records are organized hierarchically in a records management environment. A file plan differs from a taxonomy, which is intended to aid users for content search and retrieval. The purpose of a file plan is for record administrators to manage retention and disposition of records. It is used to enforce records management policies.

Note: A *business classification scheme* is a conceptual representation of the business activities that are performed by an organization. It is a hierarchical model of what an organization does. The business classification scheme is the foundation from which the records classification or file plan is developed.

A *taxonomy* is a hierarchical classification scheme to aid users in searching or retrieving content. For example, classifying music by genre can generate this list: classical, jazz, and rock. A single area, such as classical, can be further classified as concertos, sonatas, symphonies, and so on. If a user searching for a piece of music knows that it is a concerto, the user can narrow the search specifically to this category. Alternatively, if the user does not know that the piece is classical, the user needs to broaden the search.

Often, within the discipline of records management, the terms business classification scheme, file plan, and taxonomy get confused and are used to mean the same thing. In addition, there might be overlap between a taxonomy and a business classification scheme.

A file plan also enforces a security model on records through the use of access control structures. Access control determines who can view, create, or dispose of records.

In a typical business environment, different documents need to be classified into different nodes or categories within the file plan. It is sometimes helpful to think of the categories or nodes that contain the records as buckets. Each bucket is tied to a retention rule, which indicates how long records within the bucket are to be kept until they are ready for disposition. Figure 3-1 on page 63 illustrates this concept.

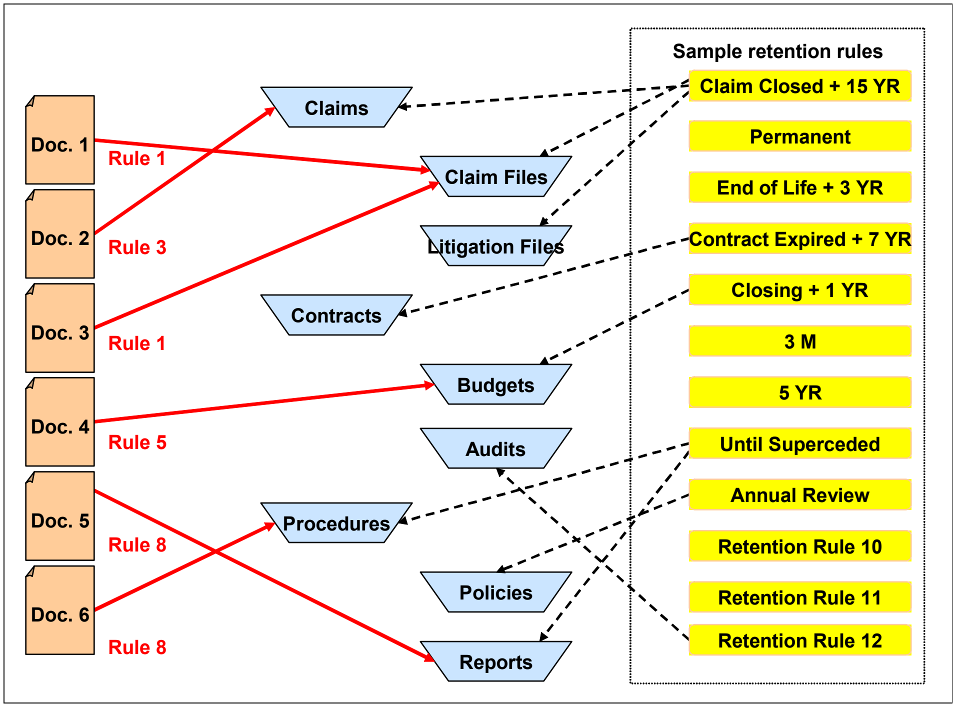


Figure 3-1 Classify a document to a file plan node that is tied to a retention rule

Note: *Classification (or cataloging)* is the act of identifying the correct file plan node or category into which a record needs to be declared. Classification places a record in context.

The term classification is not to be confused with the act of providing a security classification for a record, for instance, Top Secret and Confidential, as mandated in the DoD 5015.2 Classified standard. To mark a record as classified or to declare a classified record refers to providing a security classification marking for a record. In such cases, a record has both a file plan classification and a security classification that is independent from its file plan classification.

3.3.1 Approaches to file plan design

There are three major types of classification schemes used in file plan design:

- *Functional classification scheme*: Identifies the main functions of an organization and then breaks them down into activities and transactions. The International Organization for Standardization (ISO) 15489 Records Management standard advocates this model as the best practice for file plan design.
- *Subject classification scheme*: Focuses on the document or the record, rather than the function, which is the basis of library cataloging systems.
- *Organizational classification scheme*: Mirrors the structure of an organization. While this scheme is perhaps the simplest type of scheme to implement, it suffers from a major drawback in that organization structures can frequently change.

ISO 15489 recommends the best practice for file plan design as follows:

> *Classification by function is based on the context of a record's creation and use, rather than on the content of the record itself. This method means that the record will be classified according to why it exists – that is, its function – rather than what it is about – that is, its subject. An analysis of business activities and processes thus provides an understanding of the relationship between an organization's business and its records.*
>
> (ISO 15489, Part 2, Clause 3.2.3)

The ISO 15489 standard for Records Management advocates creating a function-based file plan, which is typically split into three levels:

- *Functions:* Represent the major responsibilities that are managed by the organization to fulfill its goals. Functions are high-level aggregates of the organization's activities. Functions are normally not aligned with organizational structures, because they are more stable than administrative units, which are often consolidated or further divided during organizational restructures. Common functions can be and often are dispersed across the structural components of an organization.
- *Activities*: Represent the major tasks performed by the organization to accomplish each of its functions. Multiple activities can be associated with each function. An activity must be based on an interrelated grouping of transactions producing a particular outcome.
- *Transactions*: Represent the smallest units of business activity. Transactions help define the scope or boundaries of activities and provide the basis for identifying the records that are required to meet the business needs of the organization. The identification of transactions also helps in the formulation of the record's description as part of an organization's retention schedule.

Best practice: When possible, use a functional classification scheme structure for file plan design by representing business functions at the highest level, business activities within those functions at next level down, and transactions within each activity at the lowest level.

Figure 3-2 illustrates the file plan structure as advocated by ISO 15489.

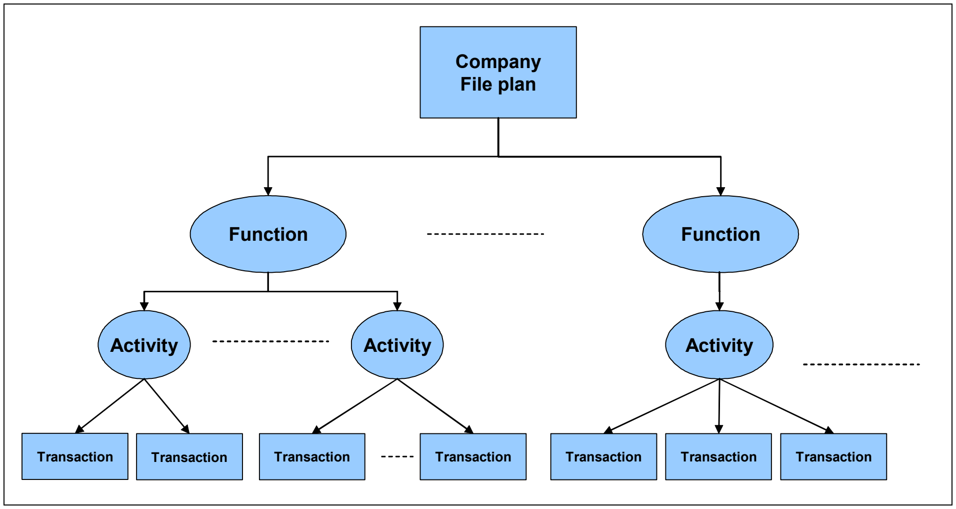


Figure 3-2 File plan structure advocated by ISO 15489

As mentioned in 3.2.4, “Conducting records inventory” on page 57, many companies have a form of record filing process. Before you design your file plan, review the record filing processes and records inventory that you generated from that section. Design a file plan that models the records and their functional relationships within the company.

3.3.2 Example file plan

Figure 3-3 on page 66 shows an example file plan structure, which we call the XYZ file plan, for a fictitious Fictional Insurance Company X, based on a functional approach to file plan design. We use this example as the framework for a more detailed case study throughout this book to illustrate various concepts related to file plan design and configuration.

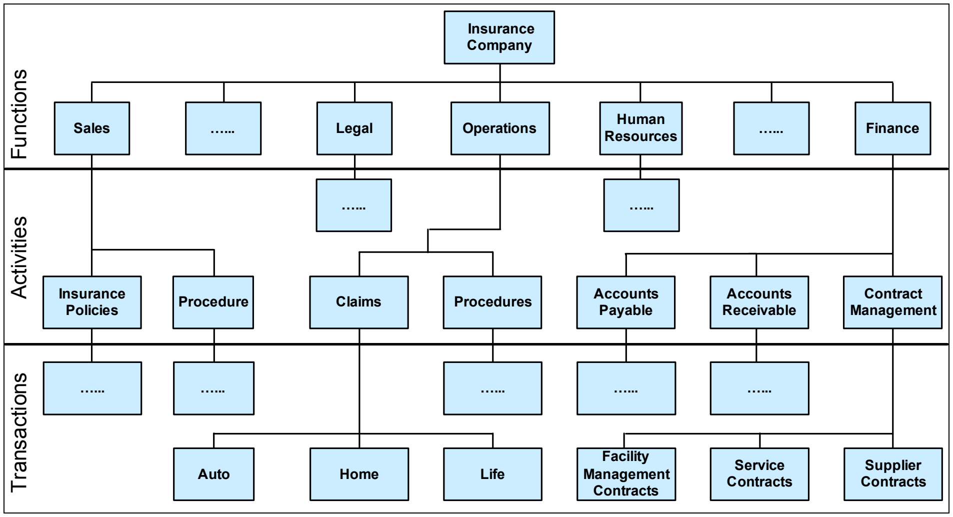


Figure 3-3 Example of a functional file plan for the Fictional Insurance Company X

This file plan has categories that represent the functions of the business, for example, Sales, Operations, and Finance. Subcategories are used to aggregate business activities within these functions. For example under Finance, the subcategories are Contract Management, Accounts Payable, and Accounts Receivable. These activities are further divided into narrower categories - transactions - that are sufficiently granular that they pertain to a relatively small subset of the overall spectrum of categories that an organization deals with on a daily basis. For example within Contract Management, there are categories to manage Facility Management Contracts, Service Contracts, and Supplier Contracts. This level of granularity is normally where a specific retention rule can apply to the records or records folders that are declared under the transaction.

Records are classified as belonging to one or more of these narrowly defined categories. This granularity allows for the efficient location of records based on the categories of information to which they belong. These categories then dictate how long records are retained. This method means that records are being retained in accordance with corporate policies rather than a user having to determine (rightly or wrongly) how long to retain a record.

While designing a file plan, it is important to determine if both electronic records and paper records will be managed within it and to ensure that any specific requirements for paper records are taken into consideration in the file plan design. For example, paper records require a different destruction process than electronic records. In this case, you might want to have separate subcategories or folders for physical and electronic records of the same type, which allows for separate disposal processes.

3.3.3 File plan elements in IBM FileNet Records Manager

In order to be able to model an organization's file plan in IBM FileNet Records Manager, it is important to understand the elements or objects that IBM FileNet Records Manager uses to build a file plan, and how they interact with each other. File plan elements in IBM FileNet Records Manager include record categories, records folders, volumes, and records. Categories, folders, and volumes serve as containers for records.

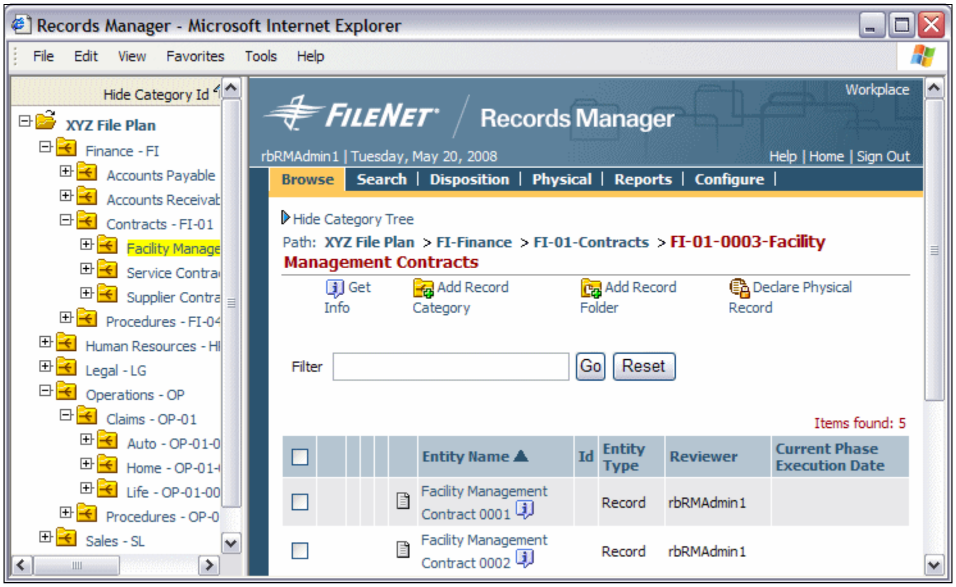
Record categories

A *record category* is a container that classifies a set of related records within a file plan. Record categories are used as the primary organizing elements to construct the tree of the file plan. You typically use record categories to classify records based on functional categories. A record category can contain subcategories or records folders, but not both. In the Base and DoD data models, you can declare records directly into categories. The PRO data model does not allow you to declare records directly into categories. By default, child entities of a record category inherit the security and disposition schedule of the record category.

A record category has a name and an ID. Both the name and the ID must be unique within the parent container.

Typically, in a function-based file plan, the functions, activities, and transactions are modeled as record categories. In the example file plan that we use in the case study (refer to Figure 3-3 on page 66), all the nodes (such as Finance, Contract Management, and Services Contracts) are modeled as record categories.

Figure 3-4 on page 68 shows the category tree from our case study with several of the nodes expanded to show the categories at the third level.



XYZ File Plan
- Finance - FI
  - Accounts Payable
  - Accounts Receivat
  - Contracts - FI-01
    - Facility Manage
    - Service Contra
    - Supplier Contra
  - Procedures - FI-04
- Human Resources - HI
- Legal - LG
- Operations - OP
  - Claims - OP-01
    - Auto - OP-01-0
    - Home - OP-01-
    - Life - OP-01-00
  - Procedures - OP-0
- Sales - SL

rbRMAdmin1 | Tuesday, May 20, 2008

Path: XYZ File Plan > FI-Finance > FI-01-Contracts > FI-01-0003-Facility Management Contracts

Items found: 5

☐					Entity Name ▲	Id	Entity Type	Reviewer	Current Phase Execution Date
☐					Facility Management Contract 0001		Record	rbRMAdmin1	
☐					Facility Management Contract 0002		Record	rbRMAdmin1	

Figure 3-4 XYZ file plan category tree and path

Figure 3-4 highlights the Service Contracts category with the following path:

```
XYZ File Plan > F1-Finance> FI-01-Contracts > FI-01-0001-Facility
Management Contracts
```

In the IBM FileNet Records Manager Web application, the path is displayed showing both the IDs and the names for each category along the path as indicated for the three categories in Table 3-3. It is a common practice to have the category ID for each lower level category include the ID from the higher parent category.

Table 3-3 Category names and IDs for the indicated path

Level	Category name	Category ID
1	Finance	FI
2	Contracts	FI-01
3	Service Contracts	FI-01-0001

Records folders

A *records folder* is a container that is used to manage and organize a group of related records that are typically disposed of together or that might need to be retained and placed on hold together as a group. For example, if you have records related to an insurance claim, it might be helpful to group all records related to the same insurance claim in the same folder. In this case, you might have thousands of records folders in the same category, one folder for each separate insurance claim.

A record folder has a name and an ID. Both the name and the ID must be unique within the parent container.

You can create electronic, physical, box, and hybrid records folders under a record category to manage electronic and physical records:

- *Electronic folder*: An electronic folder is used for storing electronic records and contains one or more volumes.
- *Physical folder*: A physical folder stores physical records and contains one or more volumes. A physical folder is a virtual entry for a paper folder. Based on your organization's physical storage structure, you can model the hierarchy of physical folders in IBM FileNet Records Manager.
- *Box*: A box is a container for physical records and provides a mechanism to model physical entities that contain other physical entities, which is in essence analogous to a cardboard box in which physical records actually are stored. A box can contain only physical records. In addition, it can contain physical folders and other box folders. A box does not use volumes. Quite often, an organization only needs to manage boxes without explicitly keeping track of the individual records or files within a box. In these cases, it is sufficient to have a description of the box contents, but the IBM FileNet Records Manager system does not have any elements representing the individual items in the box.
- *Hybrid folder*: A hybrid folder can contain both electronic and physical records and contains one or more volumes. Note that there are no behavioral differences between an electronic folder and a hybrid folder. However, a hybrid folder has additional metadata that describes a physical entity, including its home location.

Best practice: Avoid mixing electronic and physical records in the same container. Typically, physical and electronic records have different processes associated to their disposition. By keeping them separated in different containers, you can use separate disposition workflows to meet their individual disposition requirements.

Volumes

A *volume* serves as a logical subdivision of a record folder into smaller and more easily managed units. A record folder (with the exception of a box) always contains at least one volume, which is automatically created by the system when a record folder is created. Thereafter, you can create additional volumes within a record folder. Note, however, that at any given time, only one volume within each folder remains open by default, although a closed volume can manually be opened if needed. By default, the most recently created volume is open. The currently open volume is closed automatically when you add a new volume. If an automated approach is required for volume management based on a specified criteria being fulfilled, for example, a volume containing records of a specific calendar year gets closed automatically at the end of that calendar year, then custom programming is required.

A volume is the same type as its parent record folder (electronic, physical, or hybrid) and can contain the same type of records as its parent. However, a volume cannot contain a subfolder or another volume. A volume always inherits the disposition schedule of the record folder under which it is created. You cannot define a disposition schedule that is independent of the parent record folder.

Note: The concept of using a volume to subdivide the contents of a record folder comes from modeling the physical world, where you periodically need to create a new volume, because the previous volume filled up. In the context of managing electronic records, volumes have limited usefulness, but in certain circumstances, they can be used to help aggregate records based on a time interval between creating new volumes. Because creating new volumes must be managed either manually or through a custom application, there are other approaches to aggregating records that might be more suitable depending on the business requirements for disposition.

Volumes are automatically named based on the parent folder name and a sequential numbering scheme in order to uniquely identify each volume in a given folder. You cannot explicitly assign or change the name of a volume.

Records

A record object consists of a reference to the information or objects that need to be managed as a record, and it stores specific metadata about that information. A record can inherit part of its behavior from the container in which it is filed. For example, it is controlled by the disposition schedule of the parent container. For Base and DoD data models, records can be declared in record categories, records folders, and volumes. For the PRO data model, records can be declared in records folders and volumes.

IBM FileNet Records Manager supports both electronic and physical records:

- *Electronic records*: An electronic record is a record that points to an electronic document stored in the IBM FileNet Content Manager repository. You can create a separate record for each version of an electronic document or a single record for a collection of document versions (using the Declare Versions as Record option within Workplace).
- *Physical records (Markers)*: A physical record in IBM FileNet Records Manager (commonly referred to as a *marker*) is a pointer to a physical document or another object, such as paper records, tape, or microfilm, that exists in the organization. This pointer is used to store metadata about the physical object. You can store physical records in any type of record folder. However, with the exception of electronic folders, a physical record can be declared in only one container (a hybrid folder, a physical folder, or a box). That is, when a physical record is declared in a hybrid folder, physical folder, or box, you cannot file the record into another container unless the container is an electronic folder. This constraint models the physical storage for the record (for example, if you have a paper file, you can only physically store it in one box).

The constraint relationship among file plan container entities and records is illustrated in Figure 3-5.

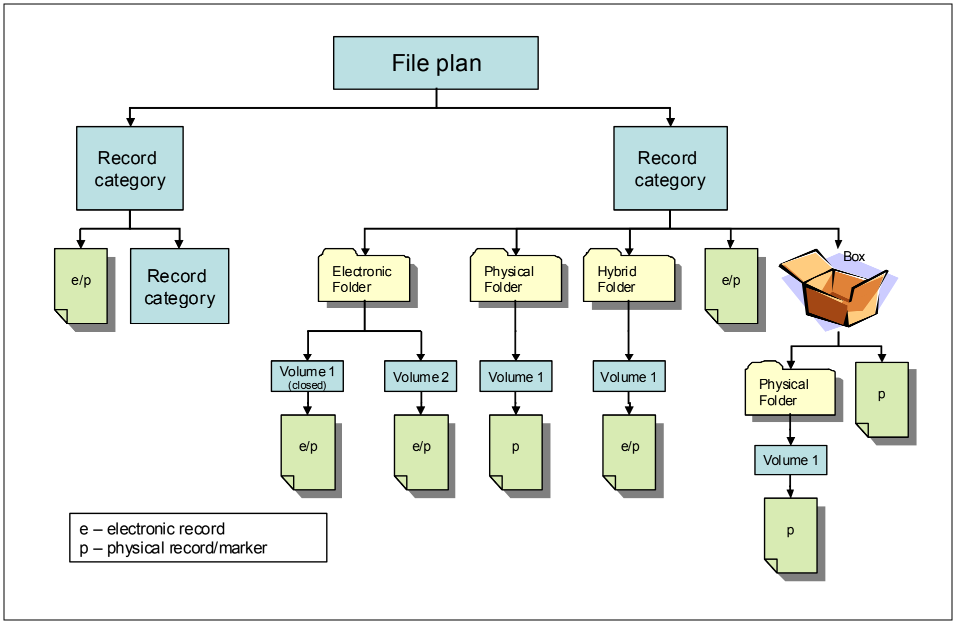


Figure 3-5 Constraint relationships among file plan elements

The file plan elements provide a flexible model in file plan design, but having this flexibility also raises the question of to when to use folders and when not to use them. The PRO data model mandates that records are always filed in folders, whereas in the Base and DoD data models, records can be filed directly in a category container without using records folders. If you need to comply with a particular standard, you must adhere to what it dictates. If you do not need to adhere to specific standards, we recommend that you use a more pragmatic approach in order to simplify the file plan design and make it more manageable. In essence, design your file plan based on the types of records that you need to declare, their relationships to other records, and how they are to be disposed.

3.3.4 Attributes of containers and records

Record categories, records folders, volumes, and records have various attributes (properties) that can be used to help identify, characterize, and organize these elements in the file plan. In addition, folders and records can be subclassed to allow for variations depending on business requirements and to facilitate the addition of custom properties that further help to organize and manage these file plan elements. In this section, we highlight a few of the common or noteworthy attributes.

Categories, folders, and records have several common predefined string properties that can be used to identify and describe each element:

- Name or title
- Unique identifier
- Description
- Subject

Physical records or boxes might have a unique bar code value as an attribute to uniquely identify each item. The system also maintains a variety of date properties, such as Date Created, Date Opened, and Date Closed, to help manage and track the elements in the file plan. These properties are just a few examples. In subsequent chapters in this book, we describe several of these properties and how they are used. For more detailed information about the attributes of containers and records, refer to the product documentation found in online help.

Vital records

Vital records are essential records needed to meet operational responsibilities during an emergency or disaster; therefore, you need to periodically review these records. To ensure periodic reviews of these records, you mark a record category, record folder, or volume as Vital, and all records created under these containers are treated as Vital.

When you mark a container as Vital, you select the recurring event that triggers the periodic review or update of vital records, and the action to launch when the review event occurs. Whenever the recurring review event occurs, the vital records' review workflow that is associated with the event is launched. IBM FileNet Records Manager provides a report, Vital Records Due for Disposal, that lists the electronic vital records due for disposition within a specific period.

Permanent records

A *permanent record* is a record that has been identified as having sufficient historical or other value to warrant continued preservation by your organization beyond the time that your organization is normally required to retain a record for administrative, legal, or fiscal purposes. It has a retention period of permanent and is identified as permanent on the records retention schedule.

You can mark a record as permanent by setting the value of its Permanent Record Indicator property to True. By default, this property does not display in the IBM FileNet Records Manager application.

You can also mark containers as permanent.

Note that there is no behavior associated with the Permanent Record Indicator property. That is, the property is informational in nature.

3.3.5 Case study: Variations in file plan design

We have developed a case study based on various types of electronic records to illustrate a variety of approaches to file plan design and configuration.

Our case study showcases three types of records: contract records, claims records, and procedure records.

Contract records (records declared into categories directly)

As shown in Figure 3-6 on page 74, the Contract Management category has three subcategories, one for each contract type: Service, Supplier, and Facility Management. We design the file plan this way, because the retention schedule might call for different contract types to have different retention periods. When a new contract document is added, it will be declared into the correct category based on the contract type. In our case study, contract records do not use any records folders to group or aggregate related records. All records are filed directly into a category and we rely on the Contract ID property to identify individual records. With this approach, you can have multiple records related to the same contract, but these records are not related by a record folder; they are related by the Contract ID.

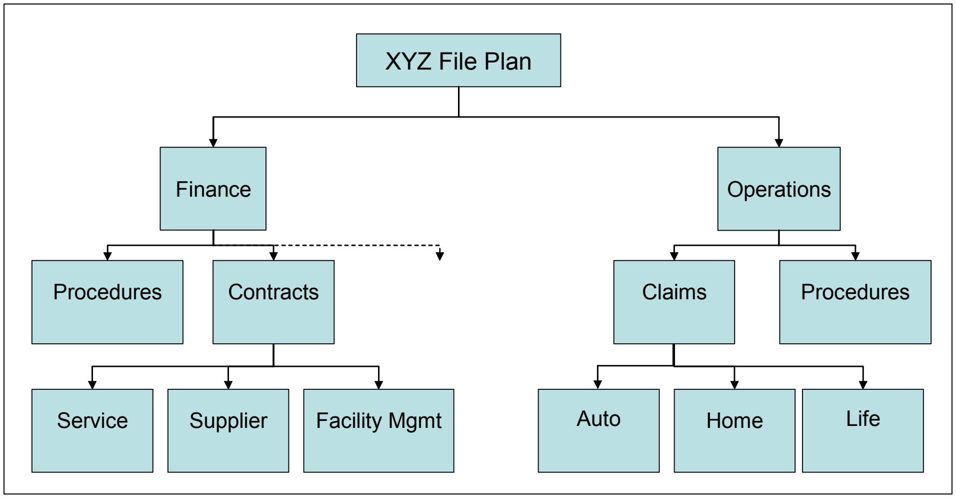
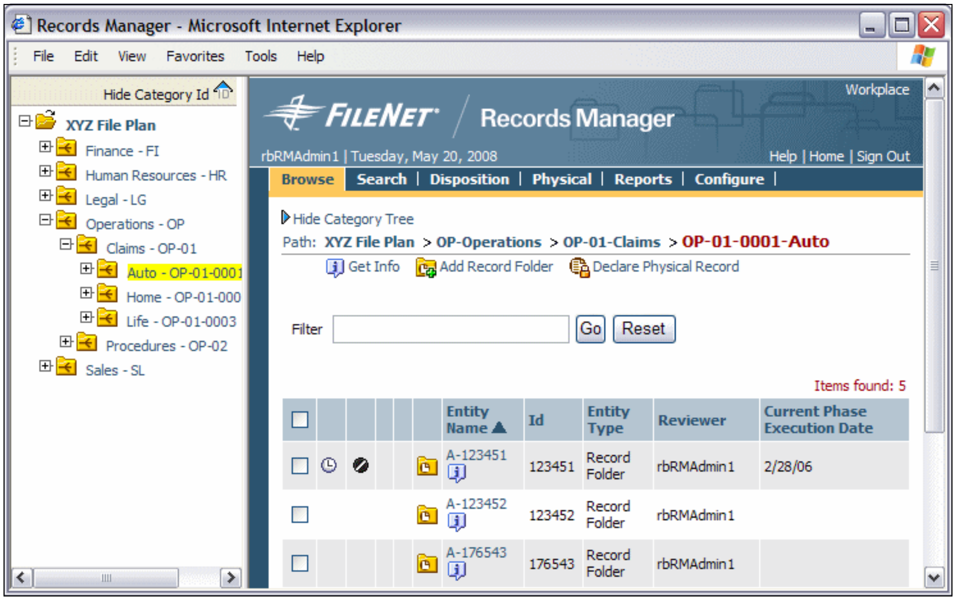


Figure 3-6 Partial file plan showing categories related to the case study for this book

In our case study, the retention schedule calls for contract records to be disposed based on the contract expiration date. As long as that date property is set correctly on each record, the records will be disposed together based on the expiration date.

Claim records (records declared into folders)

Similar to contracts, claim records are also organized by type. As shown in Figure 3-6, the Claims category has three subcategories, one for each claim type: Auto, Home, and Life. As with contracts, the retention period might vary depending on the claim type. However, for claim records, instead of declaring records directly into each subcategory, we illustrate the use of records folders to aggregate all records with the same claim number - one record folder for each claim number. When a new claim document is added, it is declared into a specific record folder based on the claim number. Figure 3-7 on page 75 shows a screen capture of this implementation with individual claim folders under the Auto category.



rbRMAdmin1 | Tuesday, May 20, 2008

Path: XYZ File Plan > OP-Operations > OP-01-Claims > OP-01-0001-Auto

Items found: 5

□					Entity Name ▲	Id	Entity Type	Reviewer	Current Phase Execution Date
□	◷	⊘			A-123451	123451	Record Folder	rbRMAdmin1	2/28/06
□					A-123452	123452	Record Folder	rbRMAdmin1	
□					A-176543	176543	Record Folder	rbRMAdmin1	

Figure 3-7 Records are declared into separate records folders for each claim number

In our case study, the retention schedule calls for claim records to be disposed based on the date of claim closure. By aggregating each claim at the claim folder level, we can rely on setting closure date for the folder and disposing of the folder along with all the records that it contains.

Procedure records

As shown in our example file plan in Figure 3-6 on page 74, we can have a category for procedures in more than one of the business units or functions. There are procedures for Operations and there are procedures for Sales. Each procedure record is declared into the appropriate category based on the business unit to which it applies. Even though these two separate categories for procedures might have the same retention rules, we separate them, because the file plan is also responsible for access control to the records. The procedure records in one business function must only be accessible to users who have permissions for that business function.

In our case study, the retention schedule calls for procedure records to be disposed based on the date superceded, which assumes that a new version of the record will be created and that it supercedes an older version.

For the remainder of this book, we use this case study to illustrate additional concepts related to file plan design, record declaration, record disposition, aggregation, record holds, and security.

The major point in this chapter is to become familiar with the various elements that make up a file plan, to understand several of the key concepts in file plan design, and to be aware that there are many options available for implementing specific business requirements. When it comes to file plan design, the business requirements are primarily driven by the retention schedule for a particular implementation.

3.3.6 Disposition schedules

Retention rules, as documented within a corporate retention schedule, are typically implemented within IBM FileNet Records Manager as disposition schedules. They determine how long a record is to be retained and how it is to be disposed at the end of its lifecycle. Disposition schedules are typically associated to a node in the file plan, resulting in records declared into, or below that node, to automatically inherit this disposition schedule.

The fundamental attributes of a disposition schedule are:

- *Disposal trigger*: A condition or event signaling that the retention period can begin

  Example: Claim documents are triggered by the insurance claim being closed; contract documents are triggered by the contract expiration; procedure documents are triggered by the date superceded.

- *Aggregation*: Level at which file plan elements will be processed for disposition

  Example: If aggregation is defined at the record level, records will be individually considered for disposition; if aggregation is defined at the record folder level, the folder (and all the records in it) will be considered for disposition as a single unit.

- *Cutoff*: An indication that the retention period and phases of disposition begin

  Example: The retention period for a contract might not begin until after a 60 day offset from the contract expiration date, allowing time to ensure that the contract has indeed expired before beginning the retention clock.

- *Retention period*: Amount of time that an item must be retained before the disposal action can be initiated

  Example: Service contracts have a seven year retention whereas facility management contracts have only a three year retention.

- *Action*: Specify what happens to the item during disposal

  Example: In many business scenarios, the most common action is to destroy records, but certain retention schedules might require records to be transferred to an external agency for permanent archiving.

We describe disposition schedules and the disposition process in more detail in Chapter 6, “Records disposition” on page 149.

3.3.7 Assigning disposition schedules to the file plan

Disposition schedules can be applied to any level of a file plan and inherited by all lower levels of the hierarchy. This inheritance can be overridden if required. In order to figure out what disposition schedules will be needed to implement the case study, it is helpful to review a partial retention schedule for the example record types that have been defined for the case study. Refer to Table 3-4.

Table 3-4 Partial retention schedule showing the case study examples

Record series	Description	Total retention period	Disposition action
Claims	Insurance claim documents (Example: claim forms, e-mail, voice recordings, regular mail, photos, and video)	Claim Closed + 5 Years	Destroy
Procedures	Internal corporate procedures for any business unit	Superceded + 10 Years	Destroy
Facility management contracts	Contract documents pertaining to facility management contracts, including the original contract, amendments, and supporting documentation	Contract Expiration + 3 Years	Destroy
Service contracts	Contract documents pertaining to service contracts	Contract Expiration + 7 Years	Destroy
Supplier contracts	Contract documents pertaining to supplier contracts	Contract Expiration + 7 Years	Destroy

Based on the retention schedule that is shown in Table 3-4, we can create the following disposition schedules and apply them to the appropriate categories in the case study example file plan:

- Claim Closed + 5 Years
- Contract Expired + 3 Years

- Contract Expired + 7 Years
- Superceded + 10 Years

How these disposition schedules are applied to the file plan is influenced by the design and structure of the file plan and the intended behavior that is being implemented.

Figure 3-8 illustrates the disposition schedules that are assigned to various categories. Considering Finance and Operations as Level 1 in the file plan, the disposition schedule *Claim Closed + 5 Years* is applied to the Claims category at Level 2, and it is inherited by all lower level categories (and the folders that are under these categories). The disposition schedule *Contract Expired + 7 Years* is applied to the Contract Management category at Level 2, and it is inherited by lower level categories. However, the disposition schedule *Contract Expired + 3 Years* is applied to the Facility Management Contracts category at Level 3 and effectively overrides any inherited disposition for that single category. The disposition schedule *Superceded + 10 Years* is applied to any category that contains procedure records.

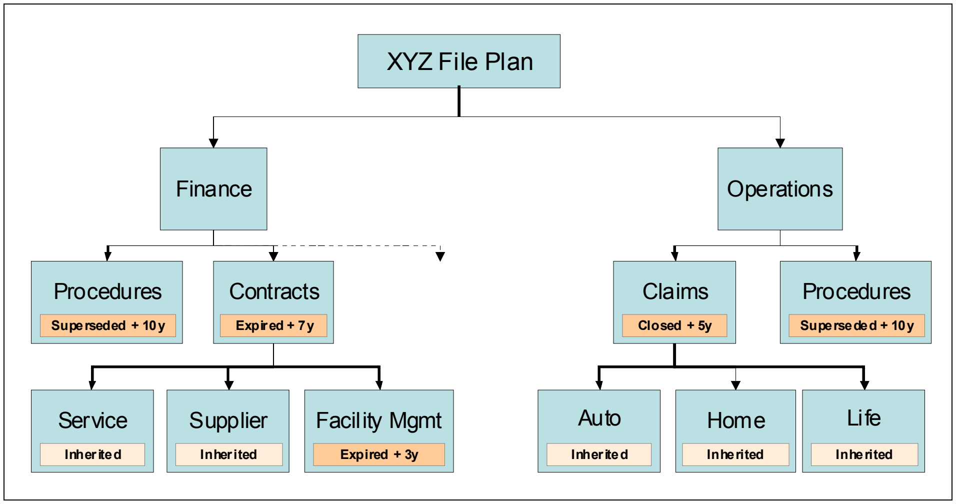


Figure 3-8 Assigning disposition schedules to categories in the file plan

3.4 Creating and maintaining a file plan

Creating and maintaining the file plan manually can be tedious work, because a file plan can often have a large number of categories. However, when first starting to configure a system to test an initial pilot design, it might be easier to create and update the file plan manually. In this section, we describe the steps that it takes to manually create a file plan and we also introduce the File Plan Import and Export Tool, which can be used for creating and maintaining larger file plans.

3.4.1 Creating a file plan manually

Creating a file plan requires the following steps:

1. Design a file plan hierarchy that is based on your retention schedule and the types of records that you need to declare. This file plan includes determining the name and ID for each category.
2. Use IBM FileNet Records Manager to create each category.
3. Use IBM FileNet Records Manager to design and create the disposition schedules.
4. Go to the appropriate categories and assign the disposition schedules accordingly.

You can create the disposition schedules first before creating each category. If you use this approach, you can assign the disposition schedules to the categories as you create them.

For information about how to create a file plan and step-by-step instructions, refer to Chapter 11, “File plan creation case study” on page 255.

Best practice: Design your file plan before you start to create the categories in IBM FileNet Records Manager. Designing the appropriate file plan to meet your specific business requirements must include a detailed analysis of your retention schedule and the record types that you intend to manage within the system.

3.4.2 File Plan Import and Export Tool

File plans can take a considerable amount of time to develop and implement and can often contain hundreds of categories. A file plan most likely will propagate through multiple systems, such as development, testing, and production systems. To ensure consistent file plan propagation and less error-prone deployment, IBM FileNet Records Manager provides a File Plan Import and Export tool. This tool is not the export manifest within an object store; rather, it is a stand-alone command line tool.

The tool and its parameters are fully documented with the online help **FileNet P8 Documentation → Expansion Products → Records Manager → File plans → How to → Use the File Plan Import and Export Tool**.

When exporting a file plan, the tool creates an output file with the file plan information in XML format. This file can then used to import the file plan into another File Plan Object Store for another system.

The following list identifies the typical components that are included in the export:

- Record categories
- Records folders
- Actions
- Disposal triggers
- Disposition schedules
- Holds (hold definitions)
- Locations

Actions, disposal triggers, disposition schedules, and holds are discussed in more detail in subsequent chapters.

If a file plan already exists in another form, it needs to be converted to the correct XML schema to then be imported into IBM FileNet Records Manager. Details about the XML schema and its elements are also covered within the online help chapter referenced earlier. When the file plan is converted into the XML schema, the tool also provides a mechanism to validate that the XML is correct before trying the import.

The tool does have a few restrictions, which are documented in the online help. We list them here for your quick reference:

- The tool does not support cross-data model export and import. For example, if you export from a PRO data model object store, you cannot import into a DoD data model object store.
- The tool does not support exporting or importing record objects, volumes, document objects, security information, and security markings.
- The tool does not support exporting conditional holds. You must run Hold Sweep to reapply these holds on entities after importing the file plan.
- There is no rollback mechanism if an error occurs.
- Importing properties with a null value does not override the existing value if one exists. For example, exporting a phase of a disposition schedule with no retention period and reimporting it into another object store where the disposition schedule already exists and a retention period is set within the phase does not update the retention period to null. However, you can manually update the XML file to update the retention period.

- Custom properties (such as choice lists) and classes must be exported and imported in a separate XML file before you import the rest of the file plan.
- After importing a file plan, entity states, such as Closed or Ready for Disposition, are no longer in effect. The purpose of the tool is to copy a file plan from one system to another. It is not a tool to migrate records or the current system state from one system to another.

3.5 File plan performance considerations

The design and subsequent implementation of a file plan impacts the performance of a system in the following areas:

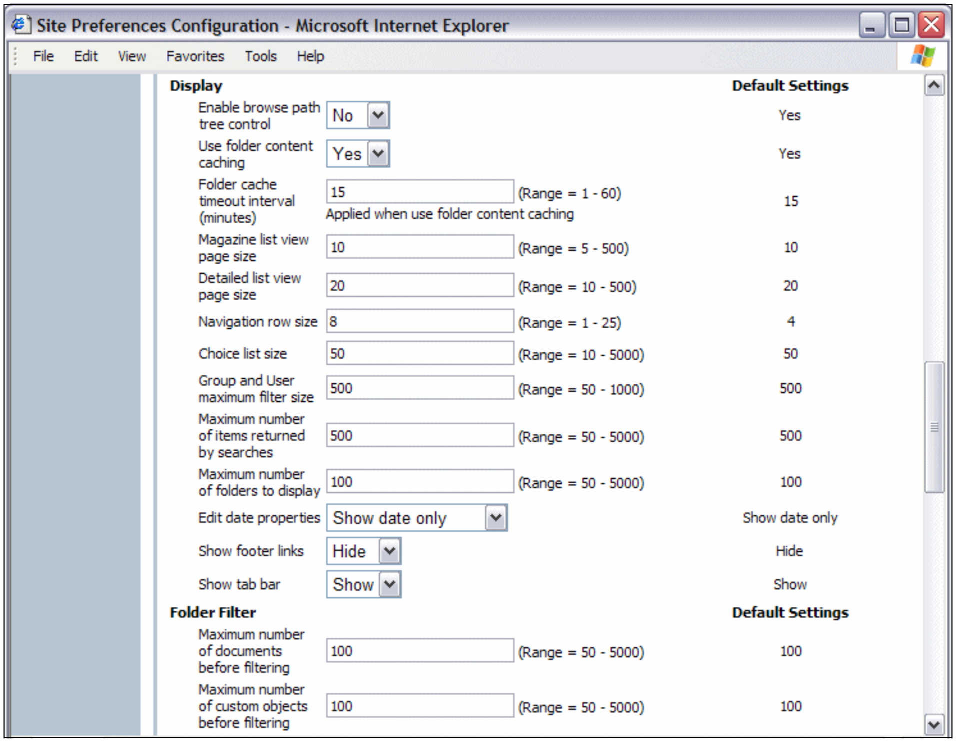
- Browsing and searching
- Disposition Sweeps and Hold Sweeps

In this section, we focus on the impact on browsing and searching the file plan. The impact of the file plan design on Disposition Sweeps and Hold Sweeps is discussed in Chapter 6, “Records disposition” on page 149.

Browsing

Browsing folders (from within the Browse tab of the IBM FileNet Records Manager application) or performing searches can return many results. In certain cases, the number of items returned is so large that the application server fails due to lack of memory. The IBM FileNet Records Manager Web application enforces a limit on the number of items that can be returned from a browse or search operation in order to greatly reduce the chance of such problems occurring.

The limits are established through Workplace Site Preferences under the General section (refer to Figure 3-9 on page 82).



Site Preferences Configuration - Microsoft Internet Explorer
File Edit View Favorites Tools Help

Display			Default Settings
Enable browse path tree control	No		Yes
Use folder content caching	Yes		Yes
Folder cache timeout interval (minutes)	15	(Range = 1 - 60) Applied when use folder content caching	15
Magazine list view page size	10	(Range = 5 - 500)	10
Detailed list view page size	20	(Range = 10 - 500)	20
Navigation row size	8	(Range = 1 - 25)	4
Choice list size	50	(Range = 10 - 5000)	50
Group and User maximum filter size	500	(Range = 50 - 1000)	500
Maximum number of items returned by searches	500	(Range = 50 - 5000)	500
Maximum number of folders to display	100	(Range = 50 - 5000)	100
Edit date properties	Show date only		Show date only
Show footer links	Hide		Hide
Show tab bar	Show		Show
Folder Filter			**Default Settings**
Maximum number of documents before filtering	100	(Range = 50 - 5000)	100
Maximum number of custom objects before filtering	100	(Range = 50 - 5000)	100

Figure 3-9 Workplace Site References page

For instance, the *Maximum number of documents before filtering* has a default value of 100, which means that only 100 documents can be returned while browsing a folder. If there are 100 000 documents filed under that folder, only 100 will be returned. A noteworthy performance point is that only 100 items will be requested from the Content Engine: that is, the Content Engine does not retrieve 100 000 items and then cut it down to 100. Content Engine constructs the queries to the underlying relational database in such a way as to limit the result set according to the established limit.

Functionally, only the established maximum is returned. Any sorting that occurs is confined to this set. Therefore, if there are 100 000 documents filed in a folder and only 100 are returned due to the maximum limit, any sorting is confined to the returned set of 100. Also, when the query is issued to retrieve the limited set, the result set is sorted by Document Title in ascending order. When sorting the result by Document Title, it appears to be accurate for the returned set. However, when sorting the result by any other field, there can appear to be missing rows because they might not have been returned in the limited set.

The maximum number of documents, custom objects, and folders that can be returned for both browsing and searches is 5 000 per object type (this is a hard limit), which means that a total of 15 000 items can be returned. This limit prevents any one user from consuming too much memory and halting the server when the user is browsing or searching through the system. Set the maximum limit carefully. If the limit is set too high and there are a number of users returning results near the limit, this situation can result in poor server performance and even a server crash.

We recommend that you do not increase these parameters beyond their defaults, not only from a server performance perspective, but it also provides a more efficient access path to records instead of a user having to page through pages and pages of results.

Based on these points, browsing only makes sense for a small file plan and a small number of records in each node. The best mechanism to access records is through searching.

The limits that we describe are only imposed in Workplace and the Records Manager application. If you are writing a custom application using IBM FileNet Records Manager APIs and performing searches through it, you must be careful to ensure that similar constraints are enforced.

Searching

If a user's main method of accessing records is through a search interface, the design of the file plan really has no impact on the search performance.

The key requirement for this method is ensuring that when defining the data model for the record or record folder classes, all the metadata that is required for searching is present. From a performance perspective, we recommend creating database indexes on the metadata elements that are used for frequent searching. For example, within the case study, searches on claims will most likely be based on the claim number or customer number (XYZClaimNumber and XYZCustomerID) and for contracts, contract ID or vendor ID (XYZContractID and XYZVendorID). These fields must be indexed accordingly.

A best practice is to only allow users to search through predefined search templates and to not allow them to create their own searches. Searches can often perform extremely resource-intensive operations on the database. Allowing users to build and run their own queries has the potential to bring the system to a halt. With search templates, system administrators know exactly how to search for records, and they can ensure that the database is tuned accordingly.

4

Security

In this chapter, we describe the IBM FileNet Records Manager security model. Security is a key component of a records management solution to control access to the records in the system, as well as to ensure that records are only deleted through the proper process.

In this chapter, we describe the following topics:

- Security model overview
- Records management roles and security
- The file plan as the primary security framework
- Individual record security
- Security and record holds

4.1 Security model overview

A key feature of a records management system is the protection of the records that are being managed. When content is declared as a record, it is no longer under the control of the author or originator. After declaration, it is under the control of the records management system based on its location in the file plan and how you choose to configure access.

By setting security on the file plan, the records manager or records administrator decides who can access the content or the metadata associated with the content. In addition, field masking can be configured to limit users to modifying only specific properties while disabling updates of other properties.

After content is declared as a record, IBM FileNet Records Manager assures that it cannot be deleted unless it is being destroyed as part of an established disposition process. Access controls are configured to prevent users from deleting either the record or content at any time, following the general best practice that records are only destroyed as part of a well-defined disposition process. Under special circumstances, certain authorized users, such as records administrators or records managers, can delete records. However, the deletion of all records is fully auditable.

IBM FileNet Records Manager leverages many of the powerful capabilities of the IBM FileNet P8 Content Engine security model. Implementing a robust security schema involves understanding how the file plan tree provides a primary structure for imposing inherited security on the records that are declared in the file plan, as well as understanding additional features of the IBM FileNet P8 security model that can be applied to records, such as security policies or marking sets. A successful IBM FileNet Records Manager security schema includes:

- Understanding the best practices surrounding access to business records
- Deciding how granular your security requirements need to be - what is provided with the product and what you have to add
- Understanding the predefined IBM FileNet Records Manager security roles and how you can use them
- Adding simple departmental or business function separation to the first level of the file plan
- Identifying typical IBM FileNet Records Manager access rights and how those might apply to departmental users (for example, the difference between a view-only role, an author role, and a coordinator role)
- Using marking sets to impose additional access restrictions on records in the file plan

As you learn about the security features that we discuss in the remainder of this chapter, remember that the Content Engine offers many more options and capabilities for modelling and controlling security than we have time to discuss in this book. Refer to the Content Engine documentation for a full discussion of all aspects of the IBM FileNet P8 security model, such as how to add and configure marking sets and how to use field masking to control access for specific properties.

4.2 Records management roles and security

When designing a records management system, whether it is manual or electronic, there are certain roles that are played by members of the organization. These roles and the actions performed by each role are typically defined by the processes and procedures of the records management system. These roles not only define the actions that can be performed by a user, but those actions that a user is specifically prohibited from performing.

IBM FileNet Records Manager has a set of predefined roles that provide a framework and starting point for defining your records management security. Each of these predefined roles has been assigned specific actions that can be performed and actions that cannot be performed by that role. The first step in creating a security model is to understand the access levels defined for each of the predefined roles. This action will help determine the ways in which you might want to further refine the security schema to meet your business requirements.

4.2.1 Four standard roles

The four standard roles defined by the Base data model are:

- Records Administrator
- Records Manager
- Records Privileged User
- Records User

These roles are typically mapped to groups in the directory service.

> **Best practice:** Avoid mapping individual users to IBM FileNet Records Manager Roles. Instead, set up and use security groups from the directory service. You can then add and remove users from these groups as needed without impacting the configuration in IBM FileNet Records Manager.

There is variation on the predefined roles with the other data models. For example, the PRO data model implements a Records Reviewer role in place of the Records Privileged User role. The DoD Classified data model implements an additional role called the Classification Guide Administrator.

It is a common practice to further differentiate the standard roles by adding specific security groups to the file plan based on business requirements. For example, you might want users from a single department to only have access to the records belonging to that department. We will discuss this topic in more detail later in this chapter. Now, we will focus on the four standard roles that come already defined in the product and how they are implemented.

We identify several of the typical responsibilities associated with the standard roles. The specific security permissions for each of these roles is configurable, and in particular, the specific variations between Records Privileged User and Records User must be determined by the needs of the business requirements that you are trying to implement.

Records Administrator

The Records Administrator is responsible for the setup and configuration of the records management system. This user is often a member of the IT department instead of the business unit or records management team. This user helps to manage the overall system, including finding and resolving issues often in conjunction with or at the request of the Records Manager. A Records Administrator works with the Records Manager to properly configure file plan components based on the business requirements. In terms of access control, a Records Administrator typically has full control of all entities defined in the file plan as well as in the File Plan Object Store (FPOS).

The Records Administrator role works primarily through IBM FileNet Enterprise Manager.

Common tasks associated with the Records Administrator role are:

- Perform initial system and component configuration
- Manage and configure security
- Configure required object classes and property templates
- Work with the the Records Manager to configure the primary file plan category structure, disposition schedules, triggers, and actions
- Import and export records
- Delete entities under special circumstances
- Configure auditing and manage audit log
- Perform backup and restore of file plan and records

- Run RecordsManagerSweep
- Perform or coordinate required database-level tasks

Records Manager

A Records Manager is typically a records management professional who makes decisions about the design of the file plan and the nature of the retention schedules to be implemented. The Records Manager works with the Records Administrator to build and maintain the file plan and all its related elements. After the system has been configured, the Records Manager is primarily responsible for monitoring the records in the system, placing records on hold, initiating disposition, and making any adjustments to the file plan and disposition schedules as business or regulatory requirements change. In terms of access control, the Records Manager typically has full control over most entities defined in the file plan, including the ability to delete records that are not on hold.

Common tasks associated with the Records Manager role are:

- Design and configure the file plan by determining required categories and folders
- Configure and maintain disposition schedules
- Allocate disposition schedules to categories or record types
- Establish holds and determine the conditions for holds based on requests from authorized business users, such as the legal team
- Place records on hold
- Initiate and approve disposition

Often, the Records Manager relies on the Records Administrator to configure the more technical elements of the system.

Records Privileged User

A Records Privileged User is a day-to-day user who typically has permissions to declare records and help manage the records in a file plan. Such a user might be a departmental records coordinator or a records office clerk. The Privileged User operates within the file plan configuration that has been implemented by the Records Manager.

Common tasks associated with the Privileged User role are:

- Create and manage records folders within a given category
- Declare records
- Update record properties
- Move records from one category to another
- Review records for disposition

Records User

A Records User is any day-to-day user who needs access to the records in the file plan but does not require the additional permissions of a Privileged User. For example, you might not allow a Records User to declare new records, but you typically allow a Records User to search for and view records.

The Records User role is the most restricted of the four standard roles. Common tasks associated with this role are:

- Search for and view electronic records
- Identify electronic documents for declaration and participate in declaring records through a well-defined business process
- Store and retrieve physical records

4.2.2 Roles and access levels

As you can see from these brief descriptions for each of the four standard roles, the one important difference among these roles is the level of access control. The Records Administrator requires the most access (is the least restricted in terms of access to records and what functions can be performed on records and other entities in the file plan) while the Records User role has the most restrictions. These four standard roles represent the most common, broad access levels required by most organizations. These access levels can be adjusted to suit specific business requirements as needed.

4.2.3 Mapping roles to security groups

It is best to establish the security groups that you want to map to the access roles before building your file plan. Security groups are mapped to access roles during the installation process or by using the security script wizard in Enterprise Manager. The security groups are established on the FPOS object store after the object store is created and the data model has been applied. Refer to the *IBM FileNet Records Manager Installation Guide* at:

http://www.ibm.com/support/docview.wss?rs=3286&context=SSNVVQ&uid=swg27010387

Knowing which security groups to map to each of the IBM FileNet Records Manager security roles is important. The most flexible approach is to create one master security group for each of the IBM FileNet Records Manager (RM) security roles. You can then later use the directory service to assign specific groups and users to the master security groups without having to change the security role mappings in IBM FileNet Records Manager.

Establishing master security groups

In Table 4-1, we illustrate four security groups where we use a naming convention that associates the groups with IBM FileNet Records Manager (P8RM) and identifies the role to which each group applies. These groups need to be established in the directory service with the intention that they will contain other groups as members. Therefore, we refer to these groups as *master security groups*. Although you can add individual users directly to these groups, you likely will add other groups, which themselves have individual users as members. The naming convention we chose here is only an example. Most organizations establish their own standards and policies for the naming of security groups.

Table 4-1 Mapping RM roles to master security groups

RM roles	Master RM security groups
Records Administrator	P8RM_RecordsAdmininstrators
Records Manager	P8RM_RecordsManagers
Records Privileged User	P8RM_RecordsPrivilegedUsers
Records User	P8RM_RecordsUsers

Best practice: Work with your security administrator to establish an appropriate naming convention for your security groups before implementing your Records Manager solution.

Using the predefined security configuration

When setting up the initial security configuration as just described with one master security group for each of the RM roles, it is possible to use the system without further differentiating groups of users. You can assign users directly to these master groups. However, most organizations have existing policies in place where directory services group memberships have already been established according to the employee’s role in the organization. In the long run, it is easier to maintain the mapping of employees into the various RM roles by assigning existing organizational groups to the master RM security groups.

Assigning existing groups to the master security groups

After you have established the master security groups, you can assign any other security groups to these master groups to give those groups access to IBM FileNet Records Manager. For example, you might already have a group called XYZ_RecordsCenterStaff. This group can be added as a member of

P8RM_RecordsManager to give all the users who are members of the Records Center full Records Manager access.

The advantage of this approach is that it allows you to adjust security on an ongoing basis without having to remap groups to roles directly in IBM FileNet Records Manager. You simply adjust security by manipulating group memberships within the directory service.

In Table 4-2, we provide an example that shows how you can manage existing security groups in your organization by establishing group membership instead of changing the direct mapping of RM Security Roles to the master security groups. In this particular example, we assume that all members of the XYZ_IT_P8Admins group will be involved as records administrators. We can easily establish a specific IT departmental group just for records administrators as well. Similarly, in this example, we assume that all of the members of XYZ_RecordsCenterStaff will be acting in the capacity of records managers. We can easily establish specific groups to break down the records center staff into subgroups, part of which serve as records managers and others who might more appropriately be privileged users.

Table 4-2 Assigning organizational groups to the master RM security groups

Master RM security groups	Existing organizational groups
P8RM_RecordsAdmininstrators	XYZ_IT_P8Admins
P8RM_RecordsManagers	XYZ_RecordsCenterStaff
P8RM_PrivilegedUsers	XYZ_Dept_1_Coordinators XYZ_Dept_2_Coordinators XYZ_Dept_3_Coordinators
P8RM_RecordsUsers	XYZ_Dept_1_Users XYZ_Dept_2_Users XYZ_Dept_3_Users

Best practice: Instead of assigning individual users to the master RM security groups, use your organization’s existing directory services groups or develop an organizational group hierarchy that reflects your unique organizational structure. These organizational groups must be independent from the IBM FileNet Records Manager roles and can be assigned to the master RM security groups as needed.

When you nest security groups, there are obviously many ways to go about arranging your organizational groups. Groups are often organized based on the functional roles of individuals within the organization. A single user can be a member of more than one organizational group, because that person might have multiple roles in the organization. The IBM FileNet P8 security model will grant the highest level of access to a user who might be in multiple groups where those groups are used to control access to the same object.

4.3 The file plan as the primary security framework

In addition to providing a structure for assigning disposition schedules as described in the previous chapter, the file plan also provides the primary structure for assigning access to the records in the file plan, which in turn controls access to the associated electronic content. The file plan takes advantage of the Content Engine's inherited security mechanism to allow Records Managers to establish access control at higher levels in the file plan, which in turn propagates to the subcategories, folders, and records at the lower levels of the file plan. In addition, more specific access controls can be applied at lower levels to allow or restrict specific individuals or groups to certain areas in the file plan.

4.3.1 Containers as security parents

IBM FileNet Records Manager establishes *record containers* (the categories and folders into which records are filed) as the security parents for the records that are contained. Hence, whenever a record is filed into a specific record folder or category in the file plan, the record will inherit the access controls specified on that container. These access controls might be applied directly to individual containers, or they might be access controls inherited from higher levels in the tree structure of the file plan. By default, there is no access control applied directly to individual records. Although it is possible to assign access controls to individual records, IBM FileNet Records Manager was designed to leverage the security parent mechanism and inherited security in order to allow for records management policies to dictate the access to records in the file plan. Whether security is inherited or applied directly, it is ultimately the security on each record that determines access. For more information about security parents and inherited security, refer to the Content Engine documentation.

4.3.2 Controlling security by proxy

When an electronic record is declared (which means it now belongs to a file plan), the security as defined by that file plan takes over. Often when an electronic document is first created or ingested into the repository (before it is declared as a record), the permissions for that document are controlled by local, departmental, or individual policies. At the time that a document is declared, the security that is determined by the file plan takes effect and completely replaces any security settings that might have been applied to the original electronic document. This file plan security is accomplished through a security proxy mechanism whereby the record object in the file plan serves as the security proxy for the electronic document that it controls in the repository. For more information about the relationship between record objects and electronic documents, refer to 5.2, “Data considerations” on page 113.

4.3.3 Relating file plan structure to access control

In certain business cases, the file plan structure corresponds to the way an organization intends to control access to the records in the file plan in addition to providing a way to organize records for disposition. However, often the way that records are organized for purposes of disposition differs significantly from the way that records need to be organized for purposes of access control. First, we examine a business case where the file plan structure provides a means of organizing records for both disposition and access.

If we take the example file plan for our case study that is described in the previous chapter, we can attempt to use this file plan structure to control access to the records in different areas of the file plan. As shown in Figure 4-1 on page 95, the categories at level 1 can be associated with security groups for each of the departments associated with the functional areas that the categories represent. In this case, all the records under Finance are accessible by members of the XYZ_Dept_1 groups, while all the records under the Operations category are only accessible by members of the XYZ_Dept_2 groups. The security assigned to a category at a particular level of the file plan can be inherited by all lower levels of the file plan, effectively controlling access to all the records in that part of the file plan. One advantage to this approach is that you can easily give a new user access to records in specific areas of the file plan by simply adding that user to the correct departmental group.

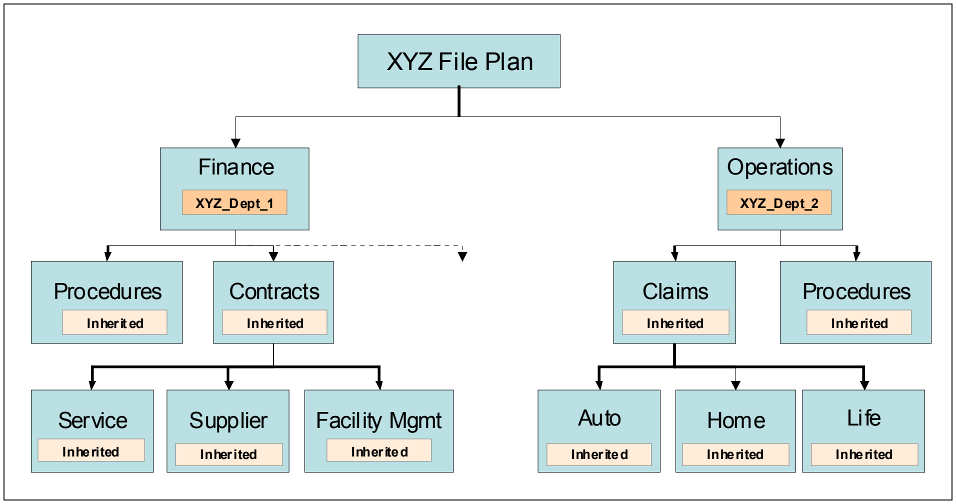


Figure 4-1 Assigning department-level security to categories in the file plan

If your security requirements call for more granular control than what is illustrated in this example, you can easily apply the same approach discussed here to lower levels in the file plan. For example, your security requirements might be better suited to a mapping of various departments to each of the level 2 categories in the file plan. In the case of our example, that means separate departments for Operations/Claims, Operations/Procedures, Finance/Contracts, and Finance/Procedures, which might make sense depending on how closely your file plan structure matches the way your users are organized.

4.3.4 Access control that differs from the file plan structure

In many business cases, the file plan structure does not correspond to the way that an organization has grouped its users. Typically, an organization can have hundreds of departments, each of which might use any area of the file plan, depending on what types of records they need to store and manage. Although a file plan structure might look like an organization chart, especially at level 1, the lower levels in the file plan often represent record types that are used by a wide spectrum of the organization. Hence, these lower level categories cannot be mapped to specific individual departments. For example, it might be the case that Contracts are primarily controlled by a specific department within the Finance area, but in reality, contracts are used and stored by almost all departments in the entire organization. Furthermore, a specific contract might be associated with a specific department even though all contracts are maintained in the file plan under Finance. In any of these cases, the file plan structure itself will not provide a good mechanism for allowing independent, more granular access, such as the access that is required for individual departments.

We can solve the requirement for independent access control in a variety of ways. In the next section, we discuss two common approaches for applying access control independently of the file plan structure: marking sets and direct security.

Best practice: Design your file plan primarily based on your retention rules and requirements and then fit your security needs into the file plan design. Use the techniques discussed here to implement a robust security schema to meet your business requirements if the tree structure of the file plan does not inherently accommodate your security requirements.

4.4 Individual record security

It is often the case that records are grouped into containers in the file plan not according to their access but according to their retention requirements. In these cases, the inherited security model that the file plan provides might not adequately address the need to control access to individual records depending upon which parts of the organization require access to these records. In this section, we describe two ways to control access to individual records, independently of the file plan structure. The first method is by using marking sets to apply a marking value to individual records. The second method is by applying direct security permissions on individual records. Each approach has advantages and disadvantages; however, each approach provides a way to satisfy requirements that call for managing the access to individual records in the file plan independently of the file plan structure.

4.4.1 Marking sets

Marking sets are defined as either hierarchical or list (non-hierarchical). For a *hierarchical marking set*, the markings are ordered from the lowest marking to the highest marking. When a user is assigned access to a marking, they also attain access to all markings lower in the hierarchy than the one to which they are assigned. The classic example of a hierarchical marking set is security classifications. In the security classification models, there is a set of markings such as Top Secret, Secret, Confidential, and Unclassified. In this model, if a user is assigned access to a Secret marking, that user has access to records that are assigned Secret, Confidential, and Unclassified. They do not have access to Top Secret records.

List (non-hierarchical) markings are not ordered; instead, each marking is independent. In this case, the user must be included in the permissions for the specific marking assigned to a record in order to access that record. List markings can used to limit access to individual records based on organizational groupings that are not represented by the file plan hierarchy, such as various departments, projects, or regions. For example, a list marking set can be used to associate a record with a specific department. The marking set is comprised of the list of all the various departments that can be assigned to a record. Users are associated with a marking by assigning a departmental group to the security permissions for the marking. After a record is assigned the appropriate marking value for its department, the marking filters access to that record, preventing any users who did not have the permissions for that department from accessing that record. A user only has access to the record if allowed by the security permissions on the marking.

Table 4-3 illustrates this example with three departments. Each department is identified by a marking value in the marking set. For each marking value, the appropriate security groups are assigned with the required access level for each group. Here, we indicate that for each department, regular users will have view only access while department coordinators will have both view and update permissions. This example illustrates that even within a single department, you can easily implement multiple access levels as long as the security schema that you design supports this approach.

Table 4-3 List marking set that defines access for individual departments

Marking value	Associated security group	Access level
Dept 1	XYZ_Dept_1_Users XYZ_Dept_1_Coordinators P8RM_RecordsManagers P8RM_RecordsAdministrators	View only View and update Full control Full control
Dept 2	XYZ_Dept_2_Users XYZ_Dept_2_Coordinators P8RM_RecordsManagers P8RM_RecordsAdministrators	View only View and update Full control Full control
Dept 3	XYZ_Dept_3_Users XYZ_Dept_3_Coordinators P8RM_RecordsManagers P8RM_RecordsAdministrators	View only View and update Full control Full control

Another aspect of the example that we provide here shows that both P8RM_RecordsManagers and P8RM_RecordsAdministrators must be included in each of the markings if you want these roles to have access to the records. Remember that marking sets only serve as filters to further restrict access to records. If we did not include these groups in the marking set configuration, users in these roles do not have access to the marked objects.

> **Note:** When referring to update permissions on records, this term means the ability to modify metadata only. After an electronic document is declared as a record, no user is allowed to modify the content associated with that record, no matter what level of access that user might have. However, it is a common business requirement that certain properties (metadata) of records are updated during the lifecycle of a record, even after it is declared. Typically, the permission to modify properties (update) is only given to a select group of users. Hence, in the example that we provide here, only department coordinators (privileged users) are allowed to update.

After the marking set has been defined as shown in Table 4-3 on page 97, you can then apply marking values to individual records in the file plan. Figure 4-2 on page 99 illustrates three individual records under the Service category, each of which is assigned a marking value (either Dept 1, Dept 2, or Dept 3) indicating the department to which it belongs. In this example, two of the records are assigned to Dept 1 and one is assigned to Dept 2. Note that by using markings, we can mix the records for various departments in a single category in the file plan, yet we still provide department-level access control. Also, note that inherited security has been applied at each of the level 1 categories to give all four master RM security groups access to the categories and records in the file plan. If you recall from the example earlier in this chapter, XYZ_Dept_1_Users was assigned to the Records Users role by membership in the P8RM_RecordsUsers master security group. Even though we specifically include XYZ_Dept_1_Users in the marking set configuration, we still need to provide these users access to the records via the file plan security.

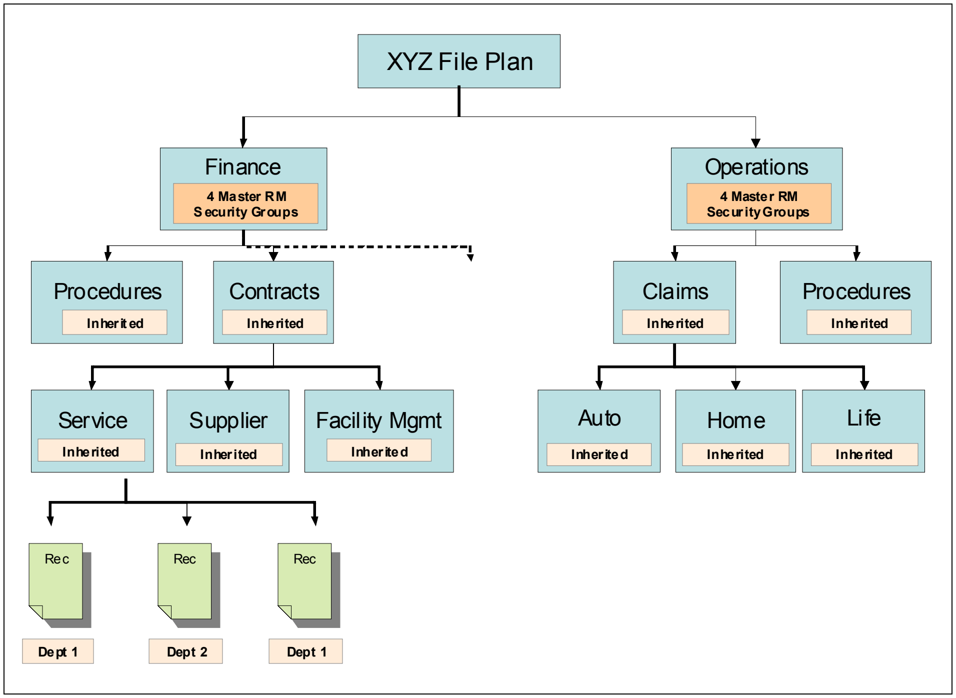


Figure 4-2 Assigning markings to individual records to control access

Marking sets are associated with the string properties of a record. By selecting a value for the marking on an individual record, marking permissions are implicitly applied to that record, which means that markings can be applied automatically when using entry templates or workflow to declare records.

When the system tries to determine whether a user can access a record, it first looks at the security permissions on the record. If the user has access based on the record's security (whether it is inherited security based on the file plan or direct security applied to each record), the system then applies any markings. The system allows access only if the user is included in the access defined by the markings. In other words, a marking filters out any users who do not have access as defined by the marking.

When using marking sets, it is important to understand that a user needs access to the marked object if you want the marking set to act as a filter. In other words, a marking set will not grant access to a record to which the user does not otherwise have access. It simply acts as a filtering mechanism for objects that a user is allowed to see using the regular security permissions, which is why we include the master IBM FileNet Records Manager security groups at level 1 in the file plan so that these permissions are inherited by the records that those users need to access.

4.4.2 Direct security

Another approach to controlling access to individual records is to use direct security on each record. This approach can achieve the same results as the use of marking sets illustrated in the previous section, but this approach requires setting security permissions directly on each record instead of simply marking each record with a marking value.

Figure 4-3 on page 101 shows our example file plan with direct security applied to individual records. Compare this figure with the one in the previous section. With this approach, we must apply security groups directly to individual records to allow those groups access to the records, instead of simply applying a marking value to each record. Notice that the records will still inherit security from the file plan, allowing both Records Managers and Records Administrators full control and full access to all parts of the file plan. However, users in any of the departmental groups will only have access to the individual records where those permissions apply. This configuration will result in the same level of access as the previous example configuration with marking sets.

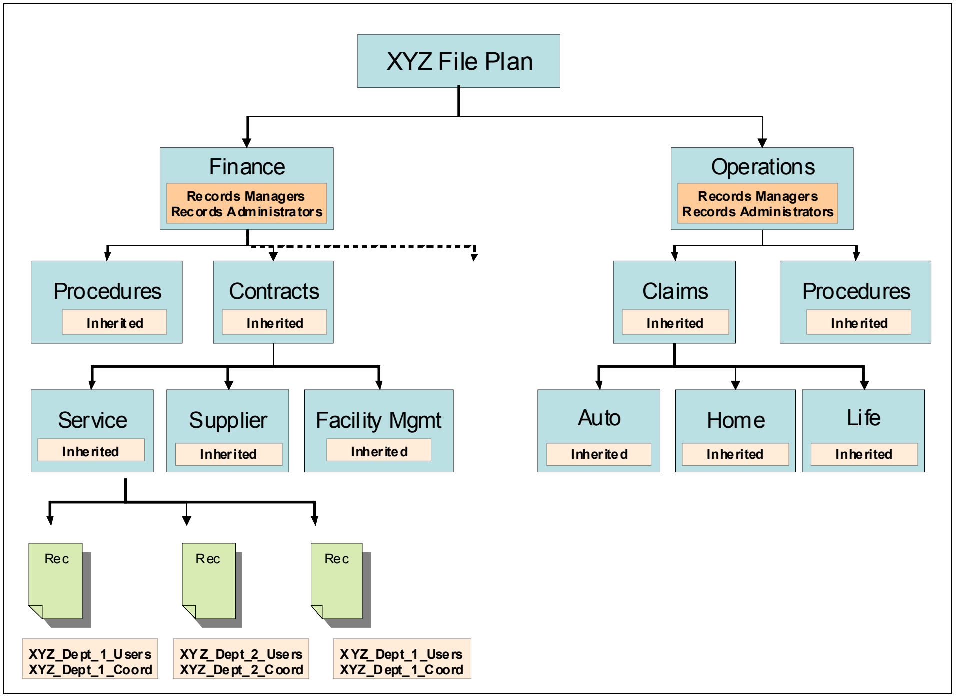


Figure 4-3 Assigning direct security permissions to individual records

4.4.3 Comparing approaches

Both marking sets and direct security allow for an ultimately flexible security schema, allowing you to control access to individual records without relying on the structure of the file plan. There are advantages and disadvantages to each of these approaches. In both cases, there is the overhead of assigning property or properties to each record.

Marking sets

The major advantage to using marking sets is being able to abstract a set of permissions into a list of marking values and to easily apply those permissions by simply setting a single marking value. The marking value can come from metadata associated with the record, such as the name of a department, that can easily be provided by a user who declares the record without that user having to understand the complexities of the underlying security schema and without having to know the exact security group names that need to be applied. The assignment of marking values can even be automated when using either entry templates or workflow for record declaration. Another advantage to marking sets is the ability to modify the permissions in one place.

With direct security, if you needed to modify the security permissions related to the group abstraction that you are using, you likely have to modify the permissions on each record. For example, imagine that you want the XYZ_Dept_1_Users to now have update permissions in addition to view content. With marking sets, you make that adjustment in one place on the marking set. With direct security, you have to make an adjustment to possibly hundreds or thousands of records.

Although marking sets provide this powerful abstraction, there is overhead required in configuring and maintaining the marking sets themselves. In addition, after you incorporate the marking set approach, you must provide a marking value for each and every record in order to provide adequate access control. As long as this approach matches your business requirements, it can be the most effective way to achieve the security configuration that you want.

Direct security

The advantage of using direct security is that you can avoid the overhead of maintaining marking sets. However, direct security is more appropriate when it can be managed and applied programmatically by a custom application that is based on business logic that is built into the application. Direct security can be cumbersome for users to apply and configure and must be avoided unless it is managed by a custom application.

> **Best practice:** Use marking sets when appropriate and feasible to apply access control on individual records. Marking sets offer the advantage of simply setting a marking value in order to apply a preconfigured set of permissions to an individual record.

4.4.4 Another example: Controlling access with markings

We used the example of separate departments requiring access control earlier in this section to introduce the idea of marking sets. Marking sets can be used to restrict access based on any conceptual groupings or divisions within an organization.

Imagine that we want to implement our example file plan for an organization that does business in several European countries. This organization wants to maintain all of its records in a single file plan, but it wants to restrict access to individual records to the country to which the record belongs. There can certainly be groups of users who have access to all records, but most users must be limited to accessing records only for the countries to which they belong.

In order to implement this example, we define a marking set with associated security groups. In Table 4-4, specific security groups for each country are associated with corresponding marking values. These marking values can then be used to filter access to specific records by country. In this example, we establish specific security groups for France, Germany, and Spain.

Table 4-4 A list marking set that defines access for individual countries

Marking value	Associated security group	Access level
France	XYZ_France P8RM_RecordsManagers P8RM_RecordsAdministrators	View and update Full control Full control
Germany	XYZ_Germany P8RM_RecordsManagers P8RM_RecordsAdministrators	View and update Full control Full control
Spain	XYZ_Spain P8RM_RecordsManagers P8RM_RecordsAdministrators	View and update Full control Full control

In Table 4-5 on page 104, organizational groups are associated with each country as dictated by business requirements. In this example, we associate specific departments with each country by simply making the organizational groups members of the specific country groups that are in turn associated with the marking set. Notice that the departmental groups are not associated directly with the marking values, but they have indirect membership by our nesting the groups appropriately. We chose to make this example a bit more complex by adding a fourth department that has been given access across two countries. In this example, Dept 1 is limited to only France, Dept 2 is limited to only Germany, and Dept 3 is limited to only Spain. However, Dept 4 has access to records marked for either France or Germany. This example illustrates the potential for flexibility and complexity available through marking sets in defining a security schema that will meet your business requirements.

Table 4-5 Assigning organizational groups to country groups

Security groups for each country	Existing organizational groups
XYZ_France	XYZ_Dept_1_Users XYZ_Dept_1_Coordinators XYZ_Dept_4_Users XYZ_Dept_4_Coordinators
XYZ_Germany	XYZ_Dept_2_Users XYZ_Dept_2_Coordinators XYZ_Dept_4_Users XYZ_Dept_4_Coordinators
XYZ_Spain	XYZ_Dept_3_Users XYZ_Dept_3_Coordinators

After establishing the marking set and the group memberships, Figure 4-4 on page 105 shows how individual records can be marked to restrict country access. In this example, Record 1 is marked for Spain, Record 2 is marked for Germany, and Record 3 is marked for France. Based on the way that the groups have been nested, Dept 1 users only have access to Record 3, Dept 2 users only have access to Record 2, and Dept 3 users only have access to Record 1. However, Dept 4 users have access to both Record 2 and Record 3.

One of the great benefits of using security groups and marking sets is that if a specific user needs access to certain records, that access can be controlled entirely by the security administrator who manages the group memberships. There is no need to modify the security on the records or the file plan.

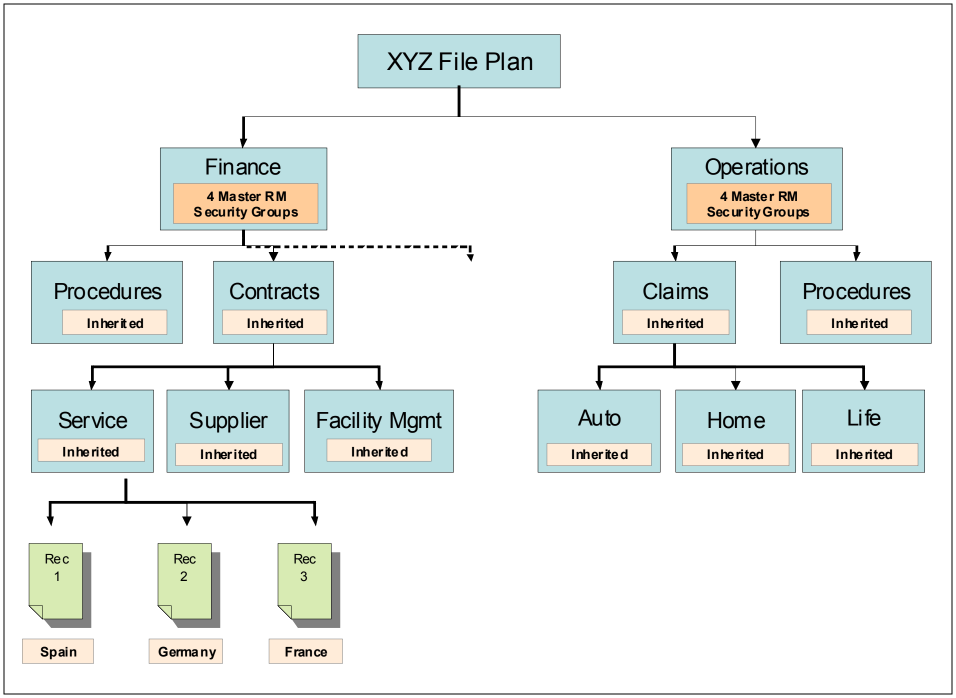


Figure 4-4 Assigning a marking to each record to control access by country

4.5 Security and record holds

No matter what access level you have, when a record is on hold, the IBM FileNet Records Manager system prevents any user from deleting or destroying that record. Even if you are a system administrator with full access control, the IBM FileNet Records Manager system is designed to prevent deletion of any record that is on hold. A record must be free of all holds before it can be deleted. Holds can be applied on individual records or on record containers (categories, folders, or volumes). Any hold that is applied to the parent container will apply to all entities (records or other containers) within the container.

For more information about record holds, refer to Chapter 7, “Records hold” on page 183.

5

Records declaration

In this chapter, we describe how to configure the system to support records declaration. We review various mechanisms available for content ingestion, how these mechanisms can be combined with a variety of record declaration mechanisms, and the benefits and constraints that come with the common choices available. We include considerations for choosing the approach that best meets your specific business requirements.

We describe the following topics in this chapter:

- Overview of content ingestion and declaration
- Data considerations
- Record classification considerations
- Primary mechanisms for ingestion and declaration
- Other mechanisms for ingestion and declaration
- Working with document versions
- Performance considerations

5.1 Overview of content ingestion and declaration

In order to design a robust IBM FileNet Records Manager system, one of the fundamental considerations is determining how new records are ingested and declared. Because the IBM FileNet Records Manager system can leverage the full capabilities of the IBM FileNet P8 Platform, there are many options available for ingesting content and declaring that content as records. The choice of ingestion and declaration mechanisms is primarily driven by customer requirements surrounding the desired user interface, the nature of the incoming records, the simplicity or sophistication of the ingestion process, and the need for a controlled business process. Because there are so many possible options for combining ingestion with declaration, we focus on several of the more common configurations and use cases.

Here are several examples:

- Customers might only need a simple document entry mechanism (such as Workplace) for users to add new electronic documents. If the documents are of specific types, customers want the system to automatically declare them as records.
- Customers might want users themselves to decide when an electronic document is declared as a record. Users have enough business knowledge to properly classify the records in the file plan when they add the documents to the system.
- Certain customers might require a simple approval process before electronic documents are declared as records. These customers might choose to use Business Process Manager (BPM) to control record declarations.
- Customers might require that the properties (metadata) from documents are validated against an external system before documents are declared as records.
- Customers might want to integrate record declaration with a larger business process - waiting for the process to complete before declaring records.

These are just a few examples of the many ways someone might choose to ingest and declare records into an IBM FileNet Records Manager file plan. Any of the various ingestion and declaration mechanisms can be mixed and matched to suit precise customer requirements.

In addition, other IBM FileNet document ingestion products, such as Capture, Email Manager, and Records Crawler[1], can be configured to support record

[1] During the writing of this book, the new product, IBM Content Collector, has been made available to the general public. It combines the capabilities of both Email Manager and Records Crawler and replaces both of these products.

declaration and provide seamless integration with IBM FileNet Records Manager.

In this chapter, we focus on describing enough detail about several of these mechanisms so that you can make the best choice for satisfying the requirements of an implementation.

Note: There are fundamental differences between declaring electronic and physical records. The scope of this chapter is limited to a discussion of electronic records. *Electronic records* have electronic content that is stored and managed in an electronic content repository.

5.1.1 Difference between ingestion and declaration

In order to make the best choices in designing and implementing an IBM FileNet Records Manager system, it is important to understand the difference between electronic content ingestion and record declaration.

Ingestion

Ingestion refers to capturing electronic content and storing it in an electronic content management system, such as an IBM FileNet P8 content repository. Electronic content can be stored without declaring it as a record in a file plan. From an IBM FileNet Records Manager perspective, this content is not under IBM FileNet Records Manager control until it has been declared a record.

Declaration

Declaration is the process of classifying content into an IBM FileNet Records Manager file plan such that the content is under IBM FileNet Records Manager control. Content can be electronic content or references to physical content.

Document as opposed to record

Many people commonly use the word record to refer to any official business document. However, from the perspective of IBM FileNet Records Manager, *records* are only those electronic documents that have been explicitly declared as records in an IBM FileNet Records Manager file plan. If a document is added to an IBM FileNet P8 content repository (ingested), but it is not yet declared as a record, we call this a *document*, not a record. As soon as the document is declared as a record, we consider the document a record.

For example, an author adds a new document to the IBM FileNet P8 content repository. The author makes several revisions to the document based on external reviews. When the document revision process is complete, the author declares the final version as a record.

Your records management policies and business requirements determine how and when you declare your documents as records.

Note: After a document is declared as a record, it is common practice in technical discussions to continue referring to the original electronic document object as a *document*. The document object is distinct from the record object. Refer to 5.2, “Data considerations” on page 113 for a more detailed discussion of document objects and record objects as they pertain to the Records-enabled content Object Store (ROS) and the File Plan Object Store (FPOS).

5.1.2 Choosing the appropriate declaration method

IBM FileNet Records Manager offers a great amount of flexibility in how you can declare records. Certain requirements call for immediate record declaration and other requirements call for delayed declaration. Certain customers require minimal user interaction while other customers need a more manual, user-intensive process. A successful IBM FileNet Records Manager implementation makes use of the most appropriate declaration method to meet business requirements.

Immediate compared to delayed declaration

Business requirements determine whether a document is declared a record as soon as it is added to a content repository or later. IBM FileNet Records Manager supports both immediate and delayed declaration as illustrated in Figure 5-1 on page 111.

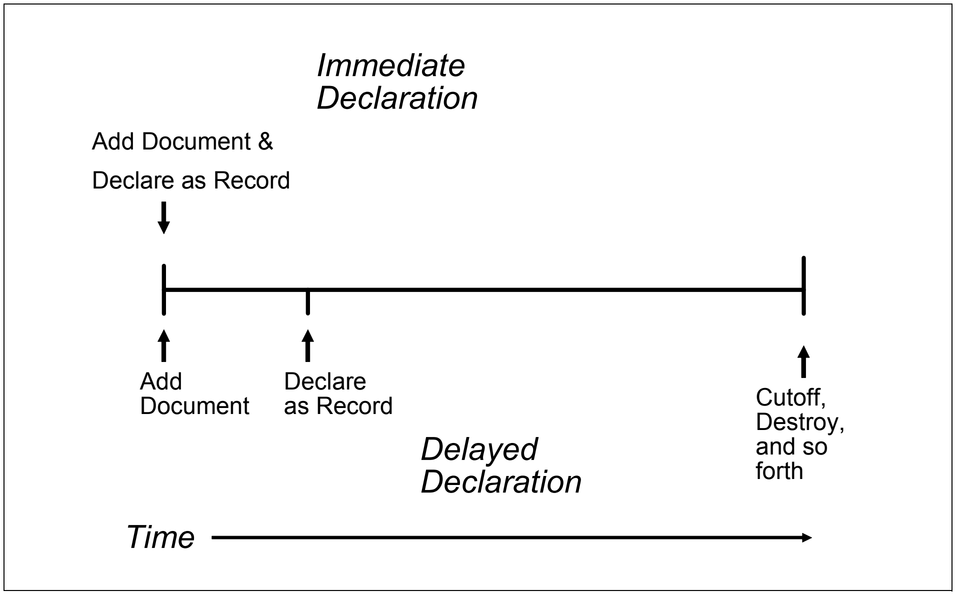


Figure 5-1 Beginning of the record lifecycle for immediate and delayed declaration

Common use cases are:

- Declare a document immediately when it is added to the content repository.
- Add a document to a repository and automatically launch an approval or verification process. Declare the document as a record only after the approval or verification process is complete.
- Declare all documents associated with a business case as records when the corresponding business process is completed.
- Declare a record only when a major version of the specified document is checked in.

Best practice: Declare a record as soon as possible given the specific business requirements. The longer you wait to declare a document as a record, the longer that document is exposed to spoliation.

ZeroClick compared to manual user intervention

The concept of *ZeroClick* promotes the idea that records declaration must use as much of the information already associated with a document as possible to automatically classify the record, requiring no additional user input.

Common use cases that leverage ZeroClick are:

- Documents are scanned into an IBM FileNet P8 content repository using IBM FileNet Capture. The scanned images are indexed and verified during the capture process and the batch of documents is committed to an IBM FileNet P8 content repository. The metadata collected during the capture process is used by IBM FileNet Capture to automatically declare the documents as records.
- Documents are dragged and dropped into folders in a file system. IBM FileNet Records Crawler is used to automatically ingest these documents into the IBM FileNet P8 content repository and declare them as records, classifying the records according to the folder structure from the file system.
- Documents are added to an IBM FileNet P8 content repository using IBM FileNet Workplace entry templates. The entry template requires the user to provide specific property values (metadata) for each document. This metadata is used by a customized Content Engine (CE) event handler to automatically classify the records.
- Documents are added to an IBM FileNet P8 content repository using IBM FileNet Workplace entry templates, with specific property values entered for each document. A workflow is automatically launched so that certain processes will be performed on the documents. When the workflow completes, the documents are automatically declared as records based on the metadata of the documents and the results from the workflow.
- E-mail messages are automatically captured by IBM FileNet Email Manager and are immediately declared as records based on specific text patterns recognized in the subject line of a message.

IBM FileNet Capture, IBM FileNet Records Crawler, and IBM FileNet Email Manager are add-on IBM FileNet P8 products that integrate directly with IBM FileNet Records Manager and provide ingestion and declaration options.

As you can see from the examples provided here, there are many ways to combine IBM FileNet Records Manager with the various document ingestion mechanisms to automate the declaration process as much as possible. However, depending on specific business requirements, specific user input might be required to properly classify the record or to approve the record for declaration. There is no single correct way to ingest and declare electronic documents as records, but there are many choices available to those people who design and implement an IBM FileNet Records Manager system.

Best practice: Achieve ZeroClick declaration by using ingestion and declaration mechanisms that take advantage of existing document metadata to automatically classify the record, thus avoiding error-prone and unnecessary user input.

5.1.3 Various user roles

For purposes of ingestion and declaration, an organization must decide on which user roles have permission to declare records. Many organizations differentiate between users who have view-only access to records and users who have permission to declare new records. Just because a user is the author of a document does not necessarily mean that the same person declares the document as a record. Again, there is no single correct way to implement a security model for purposes of declaration. The specific business requirements of the implementation must be taken into account.

5.2 Data considerations

Before you can make an informed decision on which declaration method to choose or which mechanisms are best suited for your business requirements, you must understand and configure your document and record classes and the properties associated with those classes. You need to define the various classes of documents that you intend to declare as records and define the corresponding record classes before configuring any of the mechanisms for ingestion and declaration. In addition, you might need to define specific record folder classes for the purpose of aggregating related records for disposition. For each of these classes, you need to identify the appropriate property templates that are used primarily for purposes of search and retrieval, and also for record disposition and holds.

The task of determining the appropriate document class hierarchy and associated property templates can be a large task and is required for implementing any content management system. Here, we give a brief overview of the elements that are required for a successful IBM FileNet Records Manager implementation along with several best practices. As with most issues related to system design, the choices that you make are determined by the specific business requirements that you want to implement.

5.2.1 Object stores

In order to implement the IBM FileNet Records Manager system, you need to identify at least one object store as the Records-enabled content Object Store (ROS) and one object store as the File Plan Object Store (FPOS). The ROS is where your electronic documents are stored. The FPOS is where your file plan and records-related information are stored. For more information about the ROS and FPOS, refer to 2.2.2, “Records stored as Content Engine objects” on page 39.

It is possible to implement IBM FileNet Records Manager with a variety of object store configurations. By far, the most common configuration is one ROS and one FPOS, each as a separate object store. For enterprise-level solutions, you might want multiple content repositories (requiring more than one ROS), which feed into a single File Plan Object Store (one FPOS). The primary consideration is that your content repository (document object store) must be records-enabled in order to declare any documents as records. This action is usually done during initial object store configuration and is described in detail in the *IBM FileNet Records Manager Installation Guide* at:

http://www.ibm.com/support/docview.wss?rs=3286&context=SSNVVQ&uid=swg27010387

You can also records-enable an existing Content Engine object store any time after adding IBM FileNet Records Manager to an existing IBM FileNet P8 solution.

Best practice: Your IBM FileNet Records Manager system must have at least two *separate* object stores: one Records-enabled content Object Store (ROS) and one File Plan Object Store (FPOS).

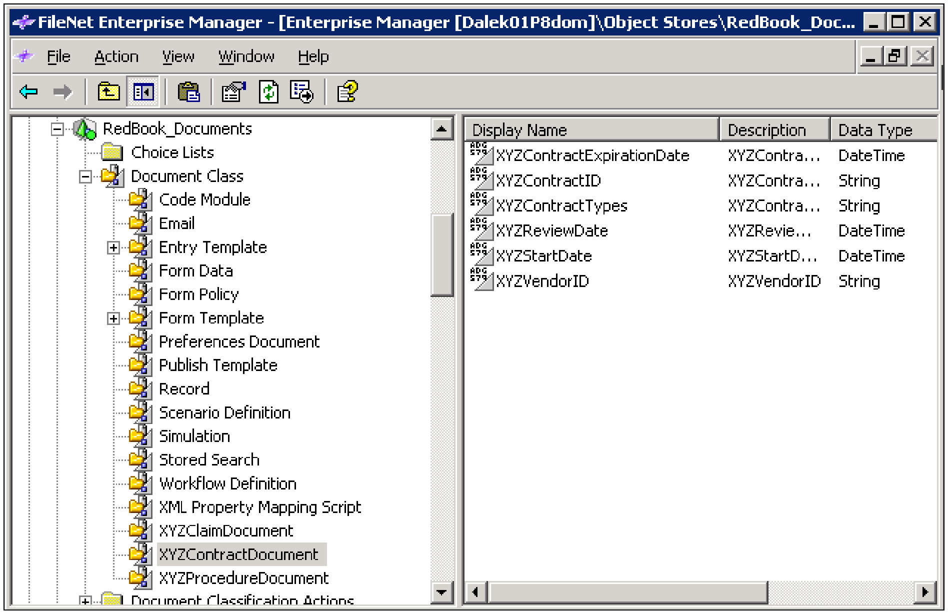
5.2.2 Document classes and record classes

After you have established your object store configuration, the next step is to determine which document classes in the ROS will be records-enabled and how those document classes map to the record classes in the FPOS.

Records-enabled document classes

Even though you can declare any document object as a record, there are many configuration-related objects in a ROS that you might not want to declare as records, such as search templates and workflow definitions. Be selective as to which classes you enable for record declaration. Figure 5-2 on page 115 shows three customer-defined document classes (those starting with XYZ) that are records-enabled. These document classes are defined in the

Redbook_Documents object store, which is the ROS for our case study. A document class is records-enabled by setting the default value of the Can Declare property to True.



Display Name	Description	Data Type
XYZContractExpirationDate	XYZContra...	DateTime
XYZContractID	XYZContra...	String
XYZContractTypes	XYZContra...	String
XYZReviewDate	XYZRevie...	DateTime
XYZStartDate	XYZStartD...	DateTime
XYZVendorID	XYZVendorID	String

Figure 5-2 Example of customer-defined document classes in the ROS

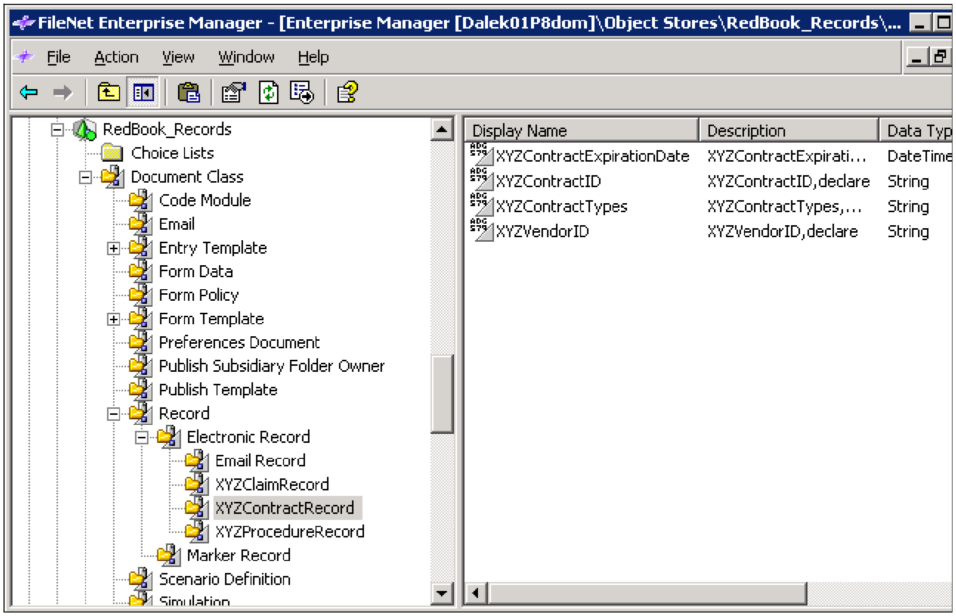
Best practice: Avoid using the root Document class as a records-enabled document class. Instead, create a subclass hierarchy from the root Document class in the ROS that identifies the classes of business documents that you will store and determine which of the document subclasses you want to enable for record-declaration. Allow users to declare documents only from these classes.

Note: IBM FileNet Records Manager does not support the declaration of CE folders or CE custom objects as records.

Record classes

After you have identified the records-enabled document classes for documents that will be declared as records, you need to create record classes in FPOS that correspond with those document classes. It is not a requirement to have a one-to-one mapping between records-enabled document classes in the ROS and record classes in the FPOS, but such a mapping is usually easier to support.

The important consideration is that both sets of classes have corresponding properties defined to support the propagation of metadata from the documents to the declared record objects. Figure 5-3 shows three customer-defined record classes in the FPOS (those record classes starting with XYZ). These record classes are defined in the Redbook_Records object store, which is the FPOS for our case study. They correspond to the three customer-defined document classes in the ROS that we show in Figure 5-2 on page 115.


FileNet Enterprise Manager - [Enterprise Manager [Dalek01P8dom]\Object Stores\RedBook_Records\...
File
Action
View
Window
Help

- RedBook_Records
  - Choice Lists
  - Document Class
    - Code Module
    - Email
    - Entry Template
    - Form Data
    - Form Policy
    - Form Template
    - Preferences Document
    - Publish Subsidiary Folder Owner
    - Publish Template
    - Record
      - Electronic Record
        - Email Record
        - XYZClaimRecord
        - XYZContractRecord
        - XYZProcedureRecord
      - Marker Record
    - Scenario Definition
    - Simulation

Display Name	Description	Data Typ
XYZContractExpirationDate	XYZContractExpirati...	DateTime
XYZContractID	XYZContractID,declare	String
XYZContractTypes	XYZContractTypes,...	String
XYZVendorID	XYZVendorID,declare	String

Figure 5-3 Example of customer-defined record classes in the FPOS

Table 5-1 shows the document classes of ROS mapped to the record classes in FPOS.

Table 5-1 Document and record class mapping in ROS and FPOS for our case study

ROS (Redbook_Documents)	FPOS (Redbook_Records)
Document Class ▸ XYZClaimDocument ▸ XYZContractDocument ▸ XYZProcedureDocument	Document Class → Record → Electronic Record ▸ XYZClaimRecord ▸ XYZContractRecord ▸ XYZProcedureRecord

Best practice: Create a subclass hierarchy under the Electronic Record class in the FPOS that corresponds to the records-enabled document class hierarchy from the ROS. This approach makes support and maintenance of these classes and properties easier.

In addition to the Electronic Record record class, IBM FileNet Records Manager supports a Marker record class for physical records and an Email record class that specifically supports properties related to e-mail.

5.2.3 Property templates

After you define the document and record classes, the next step is to determine the properties associated with these classes. The choice of properties is primarily driven by business requirements for search and retrieval, as well as requirements for managing record disposition and holds.

From a records perspective, properties are not only useful for search and retrieval, but they can be used to help classify the record during declaration. It is also common to use a custom date property for triggering records disposition. Any such properties, whether used for search or disposition, must be properly defined and configured.

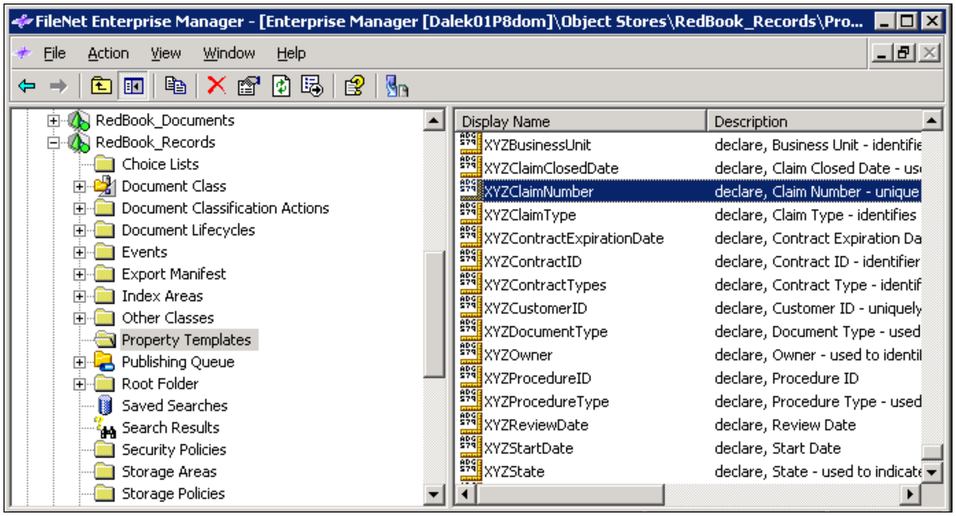
Best practice: Use the same property templates for the ROS and FPOS when possible.

We recommend the following sequence of steps to define and configure property templates:

1. Define property templates in the ROS.
2. Add the word *declare* to the Description property of each property template that you plan to use for declaration.
3. Add the properties to the appropriate records-enabled document classes in the ROS.
4. Export the property templates from the ROS using IBM FileNet Enterprise Manager.
5. Import the property templates to the FPOS using IBM FileNet Enterprise Manager.
6. Add the properties to the appropriate record classes in the FPOS.

Important: Make sure that the word *declare* is added to the property template description for any shared properties that you want to automatically propagate from the document (in the ROS) to the record object (in the FPOS) if you plan to use entry templates. If this word is missing, the entry template mechanism will not automatically propagate these property values or display these properties in the record declaration properties page of the entry template.

Figure 5-4 shows that the word *declare* is added to the Description for each property template that you want to use with Workplace entry templates for record declaration. A comma is used to separate the word from the remainder of the description text.



Display Name	Description
XYZBusinessUnit	declare, Business Unit - identifie
XYZClaimClosedDate	declare, Claim Closed Date - us
XYZClaimNumber	declare, Claim Number - unique
XYZClaimType	declare, Claim Type - identifies
XYZContractExpirationDate	declare, Contract Expiration Da
XYZContractID	declare, Contract ID - identifier
XYZContractTypes	declare, Contract Type - identif
XYZCustomerID	declare, Customer ID - uniquely
XYZDocumentType	declare, Document Type - used
XYZOwner	declare, Owner - used to identi
XYZProcedureID	declare, Procedure ID
XYZProcedureType	declare, Procedure Type - used
XYZReviewDate	declare, Review Date
XYZStartDate	declare, Start Date
XYZState	declare, State - used to indicate

Figure 5-4 Property templates with the word declare in the description

5.2.4 Document objects and record objects

Document objects are stored in the Records-enabled content Object Store (ROS) and the corresponding record objects are stored in the File Plan Object Store (FPOS). During record declaration, the original document object is kept in place in the ROS and a new record object is created in the FPOS that represents that document as a record and controls it. This record object must be added to a file plan (that is, it cannot exist in the FPOS without having the context of being filed in an IBM FileNet Records Manager container, such as a category, folder, or volume). Figure 5-5 on page 119 shows the control relationship between a record object and a document object.

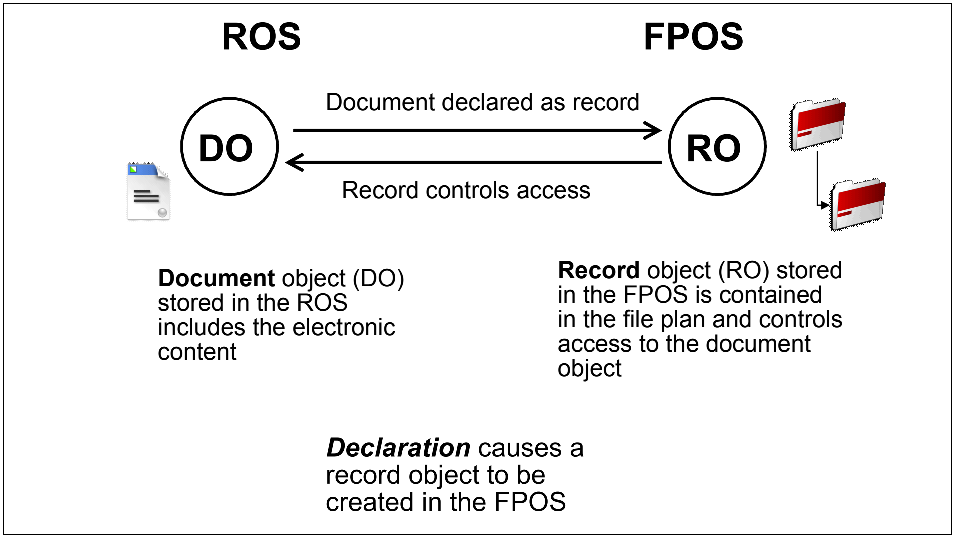

ROS
FPOS
Document declared as record
DO
RO
Record controls access
Document object (DO) stored in the ROS includes the electronic content
Record object (RO) stored in the FPOS is contained in the file plan and controls access to the document object
Declaration causes a record object to be created in the FPOS

Figure 5-5 Relationship between a document object and a record object

5.2.5 Folders

Folders can be a powerful mechanism for establishing containment relationships for stored business objects, such as electronic documents and records.

Folders in the ROS

Typically, an IBM FileNet Records Manager system does not require folders for document ingestion in the ROS. Whether to use folders in the ROS or not is an independent decision from the requirements for record declaration. Many document management strategies favor using searches for document retrieval instead of having users browse a folder structure. In such cases, folders are not needed in the ROS.

Best practice: Avoid duplicating the FPOS' file plan structure (categories and records folders) by creating a similar folder hierarchy in the ROS. There is typically no good reason for duplication. If your business requirements dictate that you have a folder structure in the ROS for organizing and managing documents, this folder structure is not a substitute for your record file plan hierarchy, nor does it necessarily have to match the file plan hierarchy.

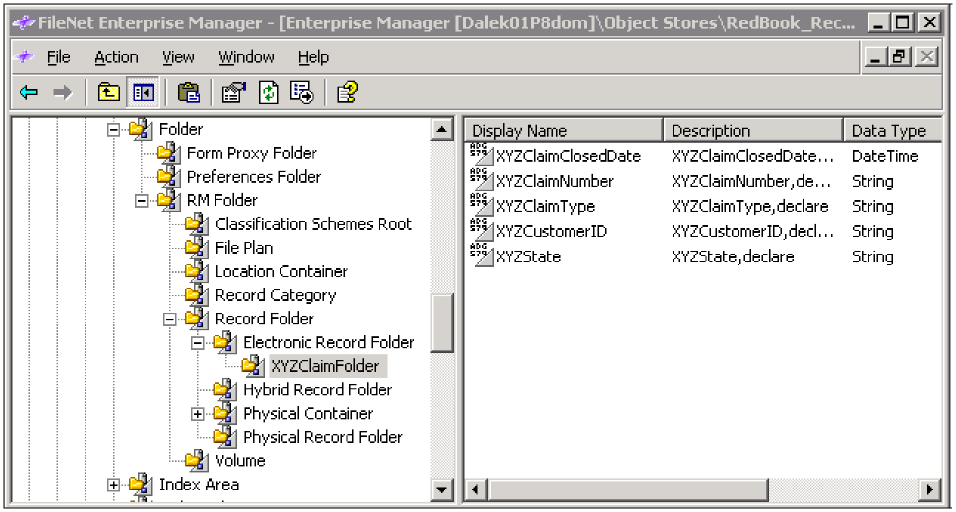
Records folders in the FPOS

Records folders can be an effective mechanism in the overall design of a file plan to allow the aggregation of related records into a common container, such as a claim folder or an employee folder, for purposes of managing records and their disposition. A record folder can be used to model a case file that serves as a container for multiple records related to a single case (such as an insurance claim, a loan, or an employment file). Records folders can also be used to group records of a similar type (such as invoices or contracts).

If you have a business requirement that calls for the use of records folders, it is usually a good idea to create a subclass of the Electronic Record Folder class in the FPOS and associate the appropriate properties with that folder class.

Best practice: If you can, design your file plan to use records folders to contain related records and aggregate for disposition at the folder level. For more information, refer to Chapter 6, “Records disposition” on page 149.

Figure 5-6 shows the XYZClaimFolder class that is used in our case study to create individual claim folders. Each claim folder contains records related to a single insurance claim. Individual records for the same claim are filed directly into the appropriate claim folder.



Display Name	Description	Data Type
XYZClaimClosedDate	XYZClaimClosedDate...	DateTime
XYZClaimNumber	XYZClaimNumber,de...	String
XYZClaimType	XYZClaimType,declare	String
XYZCustomerID	XYZCustomerID,decl...	String
XYZState	XYZState,declare	String

Figure 5-6 Customer-defined electronic record folder class in the FPOS

5.3 Record classification considerations

Having a valid file plan in place is one of the primary prerequisites for record declaration. When a record is declared in IBM FileNet Records Manager, the record must be classified (cataloged, filed, and categorized are also common terms for classified) into a file plan. Chapter 3, “Retention schedules and file plans” on page 53 describes the details of a file plan design. In this section, we highlight the importance of the file plan design as it relates to record declaration. We also show how metadata from the document can determine or control the record classification.

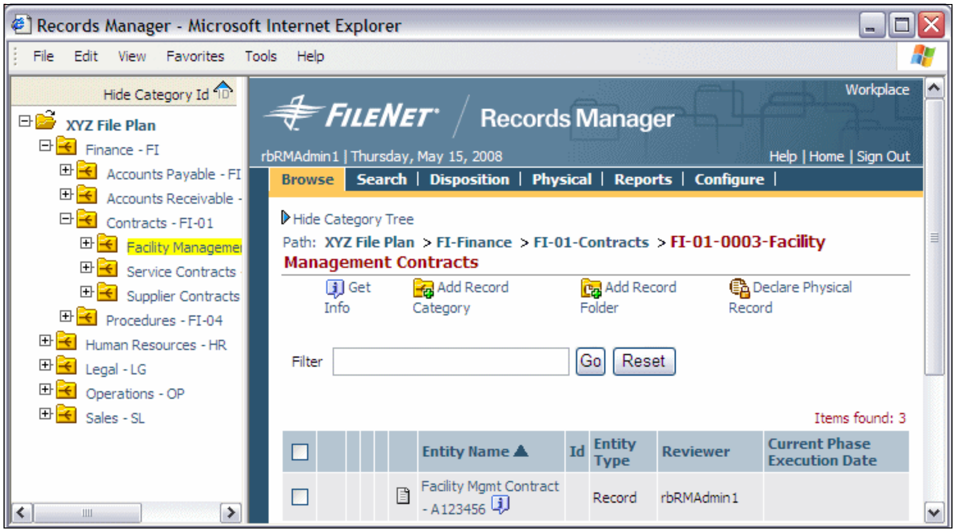
5.3.1 Identifying the parent container for a record

The key piece of information for correctly classifying a record into a file plan is the parent container for the record. This container is either a record category or a record folder.

The best way to understand how the file plan design is related to record declaration is to work with an example or case study scenario. The file plan that has been implemented for our case study showcases three examples:

- *Contract records*: They are filed directly into the appropriate record category based on the contract type. When a new contract document is added, the system must know in which category to file the record. Contract records do not use any records folders to aggregate records. All records are filed directly into a category, and we rely on the Contract ID property to identify individual records.
- *Claim records*: They are organized by claim type and make use of records folders to aggregate all records with the same Claim Number - one record folder for each claim number. When a new claim document is added, the system must know both the category and the specific record folder in which to file the record.
- *Procedure records*: They are filed directly into the appropriate category based on the business unit. When a new procedure document is added, the system must know which category to use as a parent container. One of the properties on the procedure document determines to which business unit the procedure pertains. The record is classified according to the business unit.

As shown in Figure 5-7 on page 122, Contract records are filed directly into a record category. Notice that under the Contracts category, there are three categories, one for each contract type. When a contract document is declared, the contract type determines which of the three categories is used.



Path: XYZ File Plan > FI-Finance > FI-01-Contracts > FI-01-0003-Facility Management Contracts

□	Entity Name ▲	Id	Entity Type	Reviewer	Current Phase Execution Date
□	Facility Mgmt Contract - A123456		Record	rbRMAdmin1	

Figure 5-7 Contract records are filed directly into record categories

5.3.2 File plan path

One of the easiest ways for the IBM FileNet Records Manager system to identify the parent container for a record is by providing the full file plan path for the container. The path can be represented as a string that starts with the name of the file plan and uses slashes (/) to separate each node in the path.

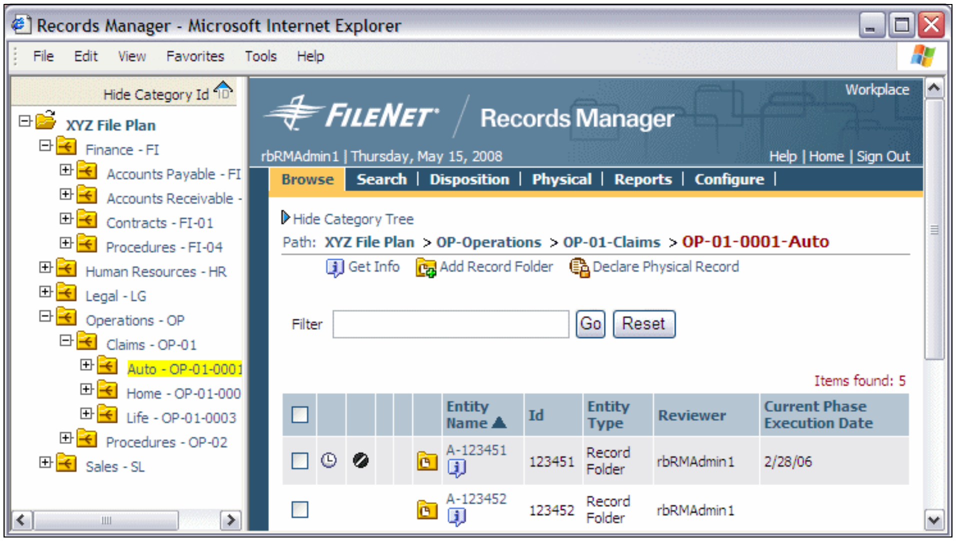
For Facility Management Contracts - A123456 (shown in Figure 5-7), its file plan path is:

```
/XYZ File Plan/Finance/Contracts/Facility Management Contracts
```

Claim records are filed in specific records folders - one folder for each claim number. In this case, both the claim type and the claim number are important properties for determining the parent container for classification.

Figure 5-8 on page 123 shows claim folders under Auto. The path for the A-123452 claim folder has the file plan path of:

```
/XYZ File Plan/Operations/Claims/Auto/A-123452
```



Path: XYZ File Plan > OP-Operations > OP-01-Claims > OP-01-0001-Auto

Items found: 5

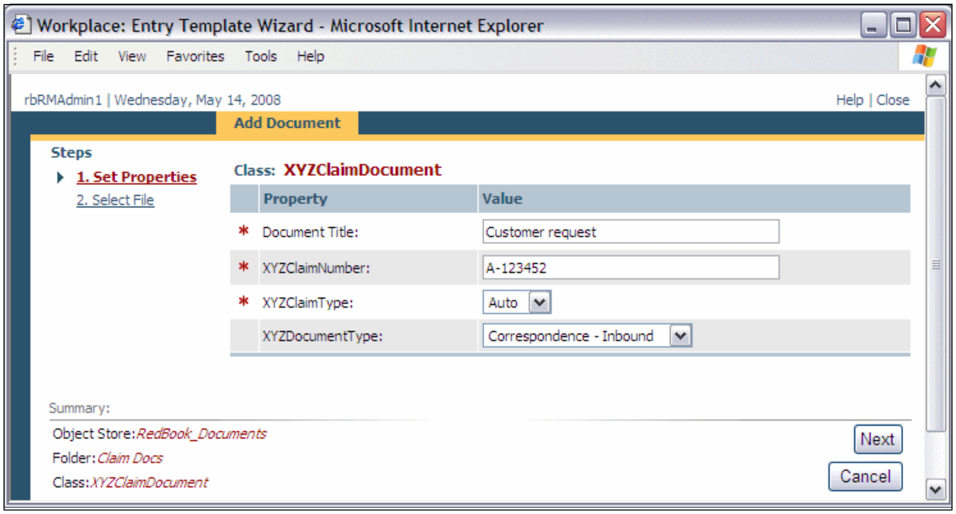
□					Entity Name ▲	Id	Entity Type	Reviewer	Current Phase Execution Date
□					A-123451	123451	Record Folder	rbRMAdmin1	2/28/06
□					A-123452	123452	Record Folder	rbRMAdmin1	

Figure 5-8 Records folders are used to aggregate claim records

5.3.3 Using metadata to determine classification

After looking at these examples of file plan paths and parent containers for record classification, let us examine the properties that might be collected for each document as the document is added to the system. The document properties can be an important source of information about how to classify the document into the file plan.

Figure 5-9 on page 124 shows the properties that a user has to enter when creating a new document of the XYZClaimDocument class.



rbRMAdmin1 | Wednesday, May 14, 2008

Class: XYZClaimDocument

	Property	Value
*	Document Title:	Customer request
*	XYZClaimNumber:	A-123452
*	XYZClaimType:	Auto
	XYZDocumentType:	Correspondence - Inbound

Summary:
Object Store: *RedBook_Documents*
Folder: *Claim Docs*
Class: *XYZClaimDocument*

Figure 5-9 Document properties can often be useful for determining record classification

Continuing with our claims example, you can see (from Figure 5-9) that XYZClaimNumber and XYZClaimType can be used to determine the file plan path for record declaration. Figure 5-10 shows how these properties can be used to construct the file plan path for record classification.

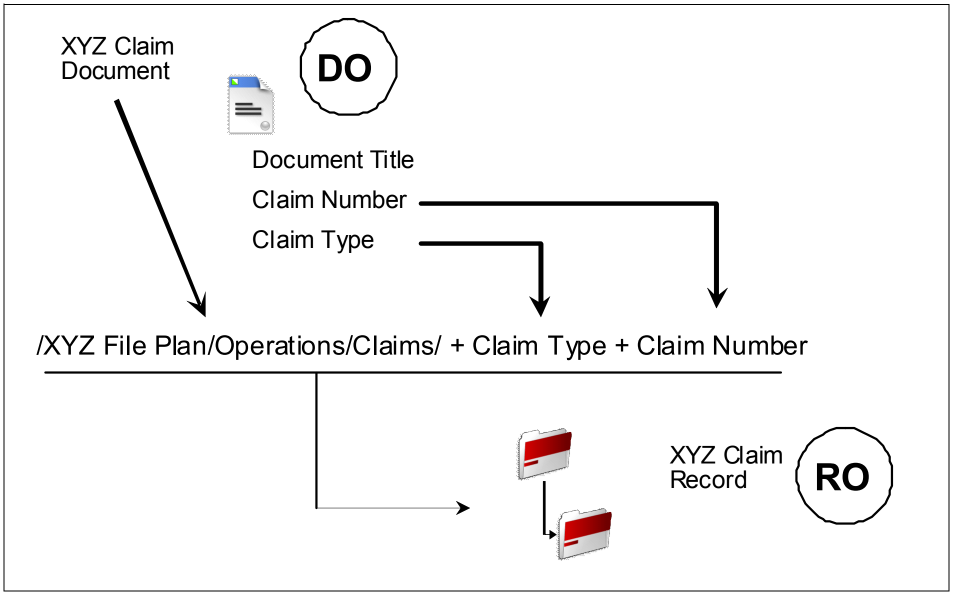


Figure 5-10 Document properties can determine the record classification

Having an understanding of how metadata (document properties) can control record classification helps in making an informed decision when choosing ingestion and declaration mechanisms to meet specific business requirements.

5.4 Primary mechanisms for ingestion and declaration

In this section, we discuss several of the common mechanisms for ingesting and declaring records:

- Workplace entry template
- Manual document entry and declaration
- Workflow with Business Process Manager
- Content Engine (CE) event handlers

These mechanisms all work with core IBM FileNet P8 Platform functionality.

5.4.1 Workplace entry templates

Workplace entry templates provide one of the fundamental interfaces for declaring electronic documents as records. Entry templates are easy to configure and require no special integration with other IBM FileNet P8 components, such as workflow or CE event handlers. Compared to manual document entry and declaration, entry templates save time by providing default information to streamline the process.

Using Workplace entry templates alone might require user intervention, depending on how the templates are configured. This section focuses on using entry templates alone, without integration with other mechanisms. Even though Workplace entry templates might be the simplest mechanism to configure, it is not necessarily the most powerful approach to automating record declaration.

Workplace with IBM FileNet Records Manager installed provides two types of entry templates for implementing a simple ingestion and declaration mechanism:

- *Document entry template*: Specifies default settings and behavior for the document entry steps (adding the new document as an author)
- *Declare as record template*: Specifies default settings and behavior for the record declaration steps.

The templates can be used alone or in combination, depending on the specific business requirements.

Document entry template

The Workplace *document entry template* allows for an automated experience by specifying default values for users and hiding certain options from them to streamline the records declaration process.

A document entry template provides the options for configuring the steps of document entry:

- Document object store and folder: Typically preselected and hidden from users.
- Document class: Typically a subclass of Document that specifies a particular customer-defined class of documents. The document class determines the metadata (properties) associated with that document and, when configured for an entry template, is typically preselected and hidden from users.
- Properties: They indicate which are required, hidden, or editable, and provide default values.
- Document security: Typically preconfigured and hidden from users. Remember that after a document is declared as a record, the record security overrides any existing document security.

For more detailed information and additional options about creating entry templates, refer to ecm_help. ecm_help is the online searchable help that is installed with IBM FileNet Content Manager, and it resides on a J2EE application server. You connect to it by pointing your browser to the server with this URL:

```
http://<host:port>/ecm_help
```

Declare as record template

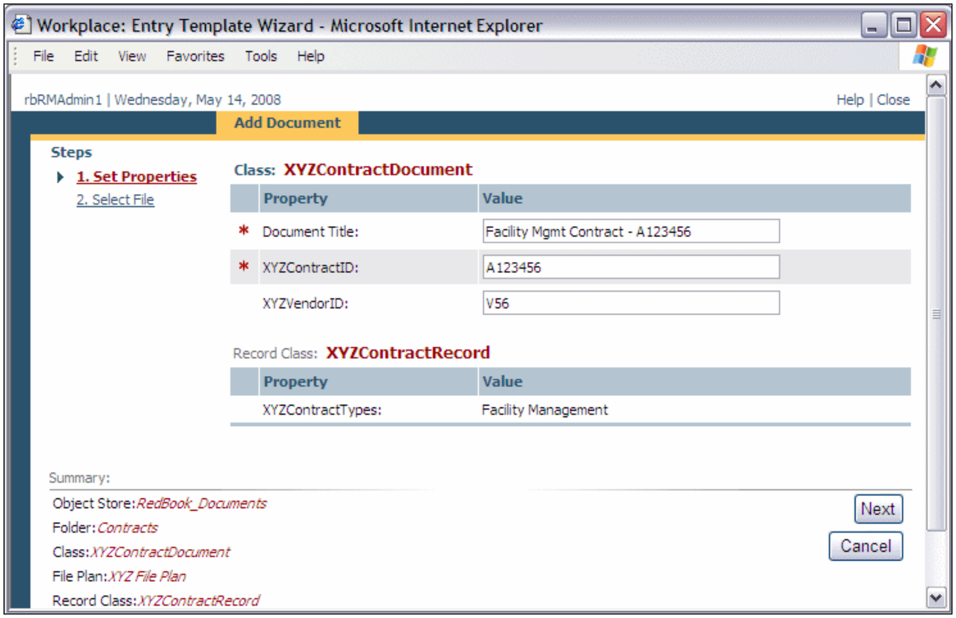
The *declare as record template* function provides the following options for configuring the steps of record declaration:

- Record class: Typically a subclass of Electronic Record that specifies a particular customer-defined class of records. The record class determines the metadata (properties) associated with that record. The selection of the record class typically is configured in the template and hidden from users during declaration.
- File plan location (record classification): The category or folder in the file plan into which to classify the record. The selection of the file plan location typically is configured in the template and hidden from the user during declaration.
- Record properties: Standard and customer-defined record properties that can be populated with metadata. The record properties are typically the metadata that is important for records management.

For more detailed information and additional options about creating entry templates, refer to ecm_help.

Combining declare as record and document entry templates

The most common configuration for ingesting and declaring electronic records utilizing entry templates is to combine a *declare as record template* with a *document entry template*. Using the two templates together provides a continuous user experience for the document author when adding a new document to the IBM FileNet P8 content repository (refer to Figure 5-11). A user adds a document, and the system immediately declares the document as a record. The user only needs to provide minimal information. All other choices and selections about how and where the record is declared and stored are hidden from the user. The necessary record declaration information is provided by the configured entry templates.



rbRMAdmin1 | Wednesday, May 14, 2008

Steps
1. Set Properties
2. Select File

Class: XYZContractDocument

	Property	Value
*	Document Title:	Facility Mgmt Contract - A123456
*	XYZContractID:	A123456
	XYZVendorID:	V56

Record Class: XYZContractRecord

	Property	Value
	XYZContractTypes:	Facility Management

Summary:
Object Store: *RedBook_Documents*
Folder: *Contracts*
Class: *XYZContractDocument*
File Plan: *XYZ File Plan*
Record Class: *XYZContractRecord*

Figure 5-11 What the user sees when adding a new document with a template

Figure 5-11 shows a single seamless user interface during document entry for the XYZContractDocument when the two entry templates are combined. The properties shown in this example are shared properties that appear in both the document and in the record. The property XYZContractID is automatically propagated from the document to the record when it is declared.

For an example of how to declare records using the combination of a *declare as record template* and a *document entry template*, refer to our case study.

Other entry template configurations

Other entry template configurations for record declaration include:

- Use a *document entry template* for adding a new document. Have a *declare as record template* that can be used later when a user wants to explicitly declare the document as a record.
- Use another mechanism for document entry and provide a *declare as record template* for users to manually declare the document as a record at a later time.
- Use a *document entry template* for adding a new document and use another mechanism, such as workflow or CE event handler, to automatically declare the record.

5.4.2 Manual document entry and declaration

Although we do not recommend using manual document entry and declaration for typical users, we describe the manual steps for document entry and record declaration. Manual document entry and record declaration can be helpful for someone who is first learning how to configure the system for future use, because it forces that person to manually step through all the different options that are available for selection and configuration. It can also be useful for advanced authors who have the knowledge to make the correct choices for all the options that are available.

Workplace document entry

Adding a new document using Workplace without the aid of an entry template requires the user to navigate through several steps, making a series of choices that help identify and store the document appropriately:

1. Navigate to a document object store and folder where you want to add a new document.
2. Select a document class and enter the associated property values for that class.
3. Set the security permissions for the document.
4. Select the document to add by browsing.

Workplace record declaration

Declaring an electronic document as a record from Workplace requires the user to navigate through several steps, making a series of choices as to how and where to declare the record:

1. Select a File Plan Object Store and record class.
2. Select the file plan location (record category or folder in which you want to classify the record) by browsing the file plan.
3. Provide additional property values required for record declaration, if needed.

> **Best practice:** Do not rely on manual document entry and declaration for business users as their routine interface with the system. Only use manual document entry and declaration from Workplace if you are an advanced author or system administrator who needs access to all available options, or if you are a system architect or application developer who wants to learn how the system works before building more automated mechanisms. After you have become familiar with your desired system configuration, choose the most appropriate ingestion and declaration mechanisms to provide as much automation as possible for users who will add documents and declare them as records.

For more options and detailed steps about how to manually add documents and declare them as records, refer to ecm_help.

5.4.3 Workflow

In many situations, entry templates alone do not provide enough flexibility to meet all of the business requirements for the record declaration process. When combined with other ingestion mechanisms, workflow can be a powerful mechanism for automating record declaration and integrating record declaration within the context of other business processes.

Flexibility of workflow

One of the most powerful features of integrating record declaration with business process workflow is the ability to design a records declaration process using the IBM FileNet BPM Process Designer tool. Such workflows can take full advantage of both Content Engine (CE) operations and IBM FileNet Records Manager operations using the component integration mechanism. Custom operations can also be integrated if required.

In addition, business process workflow can be combined with just about any document ingestion mechanism by leveraging the CE workflow subscription mechanism. No matter how a document is added to the system - whether through an entry template, IBM FileNet Capture, IBM FileNet Records Crawler, IBM FileNet Email Manager, IBM FileNet Content Federation Services - Image Services (CFS-IS), application integration, or any other ingestion mechanism - a workflow subscription can be used to automatically launch a record declaration process.

Reasons to use workflow

Because workflow is flexible, there are a variety of reasons to use it for declaring records:

- Verification or approval process to ensure that the ingested document is appropriately identified as a record
- Automatic computation of the file plan path from either the document properties or any additional properties or data that might have been collected during the declaration approval process
- Communication with external systems to either validate data before declaration or to look up data to be used in deciding how to declare the record
- Integration of record declaration with a business process where documents go through a customer-defined process before they are declared as records

Automate record declaration using workflow

Workflow can be used with any ingestion mechanism to automate record declaration by making use of the CE workflow subscription feature. A workflow subscription can be configured to trigger from a variety of CE events, the most common and useful events being either document checkin or promote version.

The following example illustrates how workflow can be combined with a simple document entry template. The workflow accomplishes two tasks that cannot be done with entry templates alone:

- Compute the file plan path based on properties from the new document.
- Introduce a simple verification or approval step into the declaration process.

The overall business process includes the following steps:

1. A user adds a new document to the ROS.
2. The system automatically launches a workflow when the document is added.
3. The workflow computes the file plan path based on the metadata from the document and identifies the correct record folder by using the path.
4. The workflow waits for a user to verify the information.

5. After the user approves, the workflow calls the declare record operation with the correct parameters.
6. The document is declared as a record.

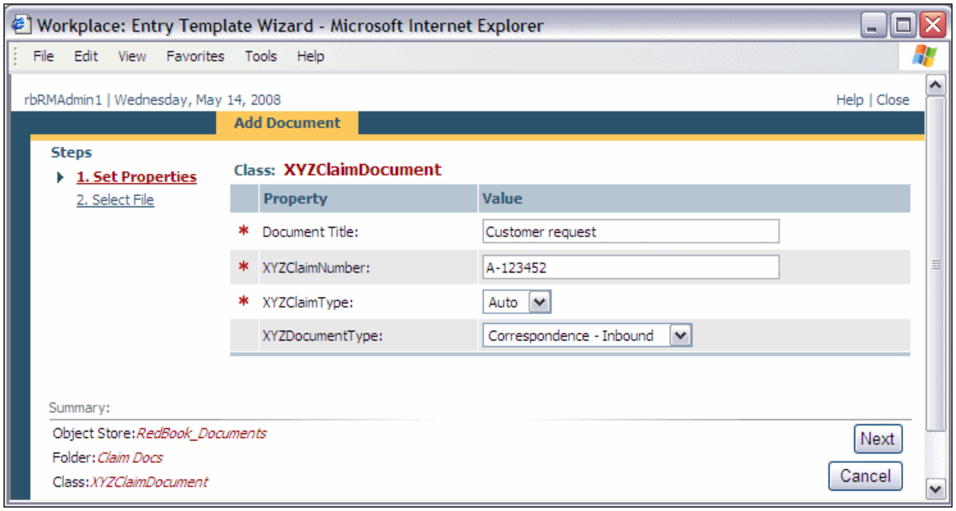
The following elements need to be in place to configure the system for this approach:

- A simple document entry template *without* record declaration enabled
- A workflow subscription that is triggered when a new document is added
- A workflow that declares the record
- A file plan that has the appropriate record categories and folders

We now look at each of these elements in more detail.

Document entry template

A simple document entry template is used to add a new document. It is *not used* in conjunction with a declare as record template. For our case study example, we are using the XYZClaimDocument as the specific document that we want to add with this process. The properties entered by the user during document entry, as shown in Figure 5-12, are used to declare the record into the correct file plan location.


Workplace: Entry Template Wizard - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Help | Close
Add Document
Steps
1. Set Properties
2. Select File
Next
Cancel

rbRMAdmin1 | Wednesday, May 14, 2008

Class: XYZClaimDocument

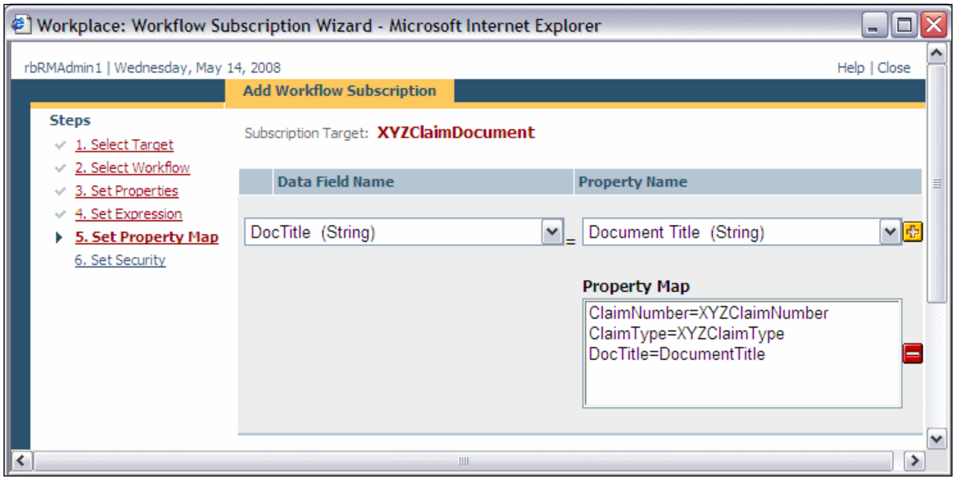
	Property	Value
*	Document Title:	Customer request
*	XYZClaimNumber:	A-123452
*	XYZClaimType:	Auto
	XYZDocumentType:	Correspondence - Inbound

Summary:
Object Store: *RedBook_Documents*
Folder: *Claim Docs*
Class: *XYZClaimDocument*

Figure 5-12 Adding document using a document entry template

Workflow subscription

A *workflow subscription* launches the workflow when a new document of a given class is added to the object store. The workflow subscription must be configured for a specific document class, in this case, XYZClaimDocument. The subscription includes a property map for specific document properties to be included for use in the workflow as shown in Figure 5-13.


Workplace: Workflow Subscription Wizard - Microsoft Internet Explorer
rbRMAdmin1 | Wednesday, May 14, 2008
Help | Close
Add Workflow Subscription
Steps
1. Select Target
2. Select Workflow
3. Set Properties
4. Set Expression
5. Set Property Map
6. Set Security

Subscription Target: XYZClaimDocument

Data Field Name	Property Name
DocTitle (String)	Document Title (String)

Property Map

ClaimNumber=XYZClaimNumber
ClaimType=XYZClaimType
DocTitle=DocumentTitle

Figure 5-13 Workflow subscription property mapping

Workflow

The workflow is designed to calculate the file plan path based on property values that are entered during document entry. It also retrieves the appropriate IBM FileNet Records Manager Container where you want the record to be declared. Refer to Figure 5-14 on page 133.

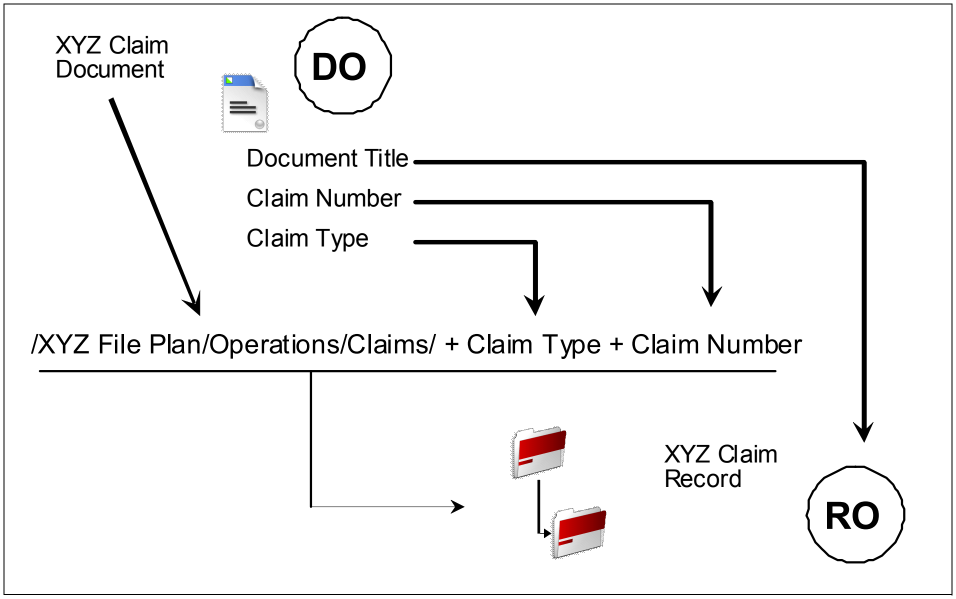


Figure 5-14   Document properties determine record classification and properties

The sequence of the steps is:

1. The workflow is launched from a workflow subscription, and relevant properties are mapped from the document so that they can be used in the workflow.
2. The workflow determines a specific claim folder and category in the file plan based on the claim number and claim type that are provided for the document.
3. The workflow includes a verification step so that a specific user other than the one who added the document verifies that the record is ready to be declared.
4. The workflow calls the declare record operation and propagates property values from the document to the record being declared.

The workflow process map in Figure 5-15 on page 134 shows these steps as they have been implemented for the example in our case study and the details of the Declare step.

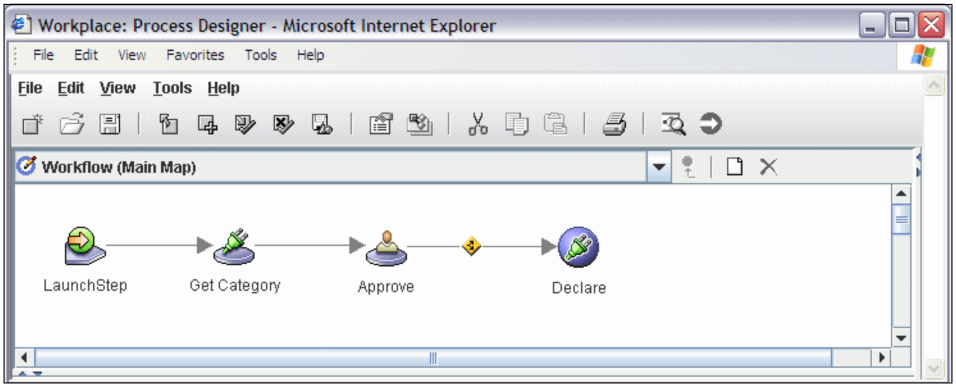
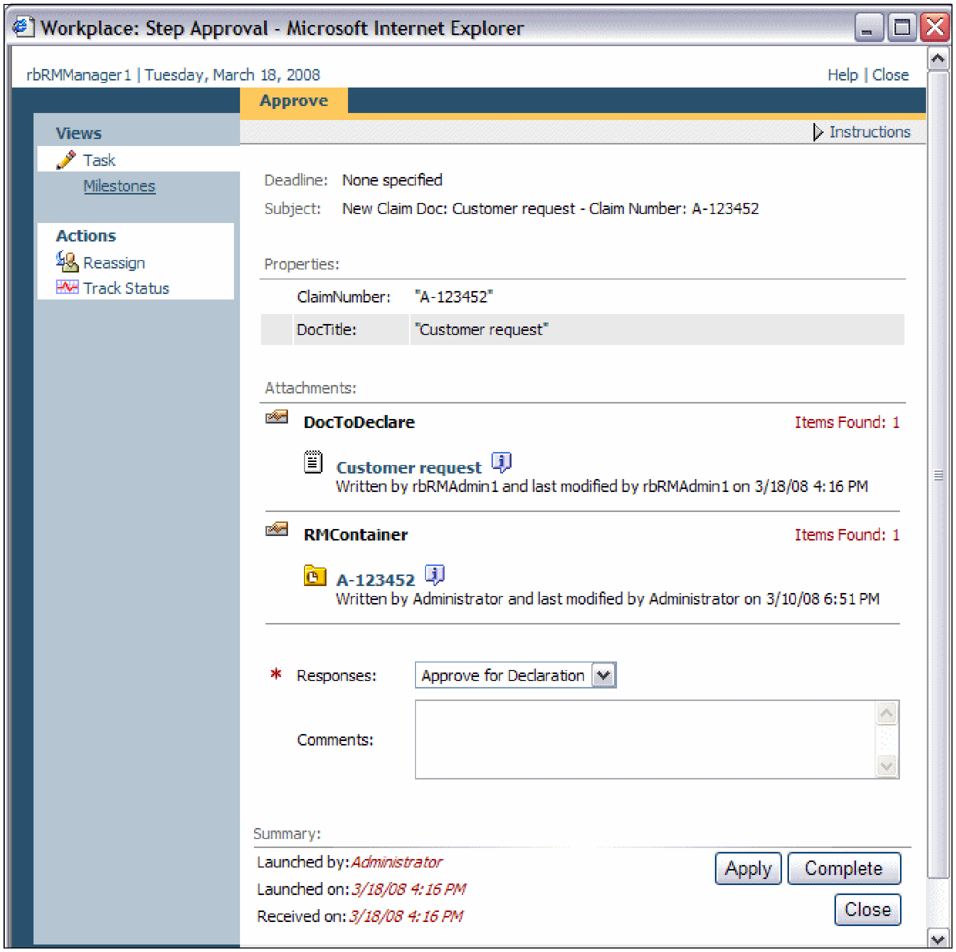

Workplace: Process Designer - Microsoft Internet Explorer
File Edit View Favorites Tools Help
File Edit View Tools Help
Workflow (Main Map)
LaunchStep
Get Category
Approve
Declare

Figure 5-15 Workflow retrieves the correct container and declares the record

Workflow process maps can be configured using the Workplace Process Designer. This example shows a simple verification process. The Declare (record operation) step can be included in any custom workflow, thereby allowing record declaration to be integrated into a larger business process depending on your business requirements.

Figure 5-16 on page 135 shows the Approve step processor (the user interface for completing the Approve step). Workflow step processors can be configured and customized to meet specific requirements for user interaction. This particular step processor makes the document available as an attachment and includes a simple pull-down list of responses for the user to choose before completing the step.


Workplace: Step Approval - Microsoft Internet Explorer

rbRMManager1 | Tuesday, March 18, 2008

Deadline: None specified

Subject: New Claim Doc: Customer request - Claim Number: A-123452

Properties:

ClaimNumber:	"A-123452"
DocTitle:	"Customer request"

Attachments:

DocToDeclare — Items Found: 1

Customer request
Written by rbRMAdmin1 and last modified by rbRMAdmin1 on 3/18/08 4:16 PM

RMContainer — Items Found: 1

A-123452
Written by Administrator and last modified by Administrator on 3/10/08 6:51 PM

* Responses: Approve for Declaration

Comments:

Summary:

Launched by: *Administrator*

Launched on: *3/18/08 4:16 PM*

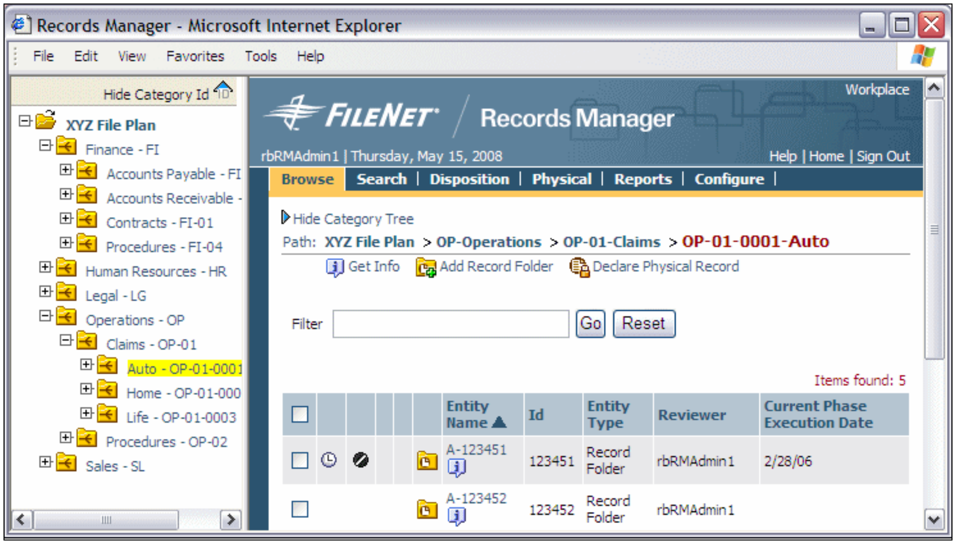
Received on: *3/18/08 4:16 PM*

Figure 5-16 Approval step before the record is declared

File plan

A file plan with the appropriate IBM FileNet Records Manager containers is already in place to receive the declared records. In this case, we need to have a valid Claim Folder in the file plan for each claim number that we intend to use. The file plan supports the declaration of multiple claim documents for each claim number. All claim documents for a given claim are filed in the same claim folder. In Figure 5-17 on page 136, our example shows several claim folders in the auto claims category (Claims → Auto). In an actual implementation, the category for auto claims might contain thousands of claim folders.

Note: In an actual implementation, you might have thousands of auto claim folders, one for each auto claim in the file plan. As long as users are not browsing the file plan, but they are using searches to find records and folders, a single file plan container has no hard limit on how many items it can contain.



rbRMAdmin1 | Thursday, May 15, 2008

Path: XYZ File Plan > OP-Operations > OP-01-Claims > OP-01-0001-Auto

Items found: 5

□					Entity Name ▲	Id	Entity Type	Reviewer	Current Phase Execution Date
□	◷	●			A-123451	123451	Record Folder	rbRMAdmin1	2/28/06
□					A-123452	123452	Record Folder	rbRMAdmin1	

Figure 5-17 Records are declared into the correct claim folder

5.4.4 CE event handlers

In certain situations, using the CE event handler mechanism to automate records declaration is more suitable than using workflow. CE event handlers enable you to directly execute Java code to perform record declaration. The event handler is triggered by a CE subscription.

CE event handlers might be appropriate in the following situations:

- You do not require a user step to verify or approve the record declaration, and you have no other reason to use workflow.
- You are using an ingestion mechanism that does not directly support record declaration, such as IBM FileNet Content Federated Services - IBM FileNet Image Services (CFS-IS).
- You can provide all of the record information including the record class and full file plan path as properties on each document that is ingested for declaration.
- You have the resources to modify or write Java code for customization.

Similar to workflow, CE event handlers can be used to construct the file plan path for record classification, which is a task that you cannot achieve with entry templates alone. In addition, you can directly modify and enhance the event handler by writing Java code to perform any number of functions before or after record declaration. CE event handlers offer the ultimate flexibility, but they require the overhead of maintaining Java code.

Examples of what you can do from a CE event handler include:

- Use document properties to look up additional information in an external system before record declaration and use the data from the external lookup to populate certain record properties.
- Compute the file plan path from the document properties (or other values that have been looked up in an external system).
- Update external systems before or after record declaration.
- Update document properties after record declaration.
- Find the previous version of a document that has been declared as a record and supercede the older record.

Similar to workflow, a CE event handler can be combined with any ingestion mechanism that adds documents to a records-enabled IBM FileNet P8 content repository (ROS). The CE event handler uses the CE event action and subscription mechanism to execute the event handler when the appropriate event (such as checkin) happens to a document in the ROS.

Autodeclare event handler

IBM FileNet Records Manager provides sample source code and .jar files for automatically declaring a record using a CE event handler. The sample source code and .jar files are a good starting point for implementing a custom event handler. The autodeclare event handler can be used to trigger record declaration if the source document includes properties that indicate the record class and the file plan path.

You can implement the autodeclare event handler using the following steps:

1. Define two string property templates in the ROS:
   - One for storing the name of a record class
   - One for storing the full file plan path that you want to use for record classification of a given document
2. Add these two properties to a records-enabled document class in the ROS. Making these properties required properties will prevent errors during record declaration if their information is missing.

3. Modify the configuration (.properties) file to reference the two properties just described as well as the correct FPOS where you intend to declare the records.
4. Define an event subscription on the records-enabled document class that will trigger the autodeclare event action based on the document checkin event.

When the autodeclare event handler is implemented, you can test it by adding a new document to the ROS using any ingestion mechanism. You must provide the record class and the full file plan path for classification in the appropriate string properties (or you can set default values for these properties and hide them from the user who is adding the document).

The autodeclare event handler will declare the document as a record based on the values that you provide on the document for the record class and the full file plan path.

Limitations of the autodeclare event handler

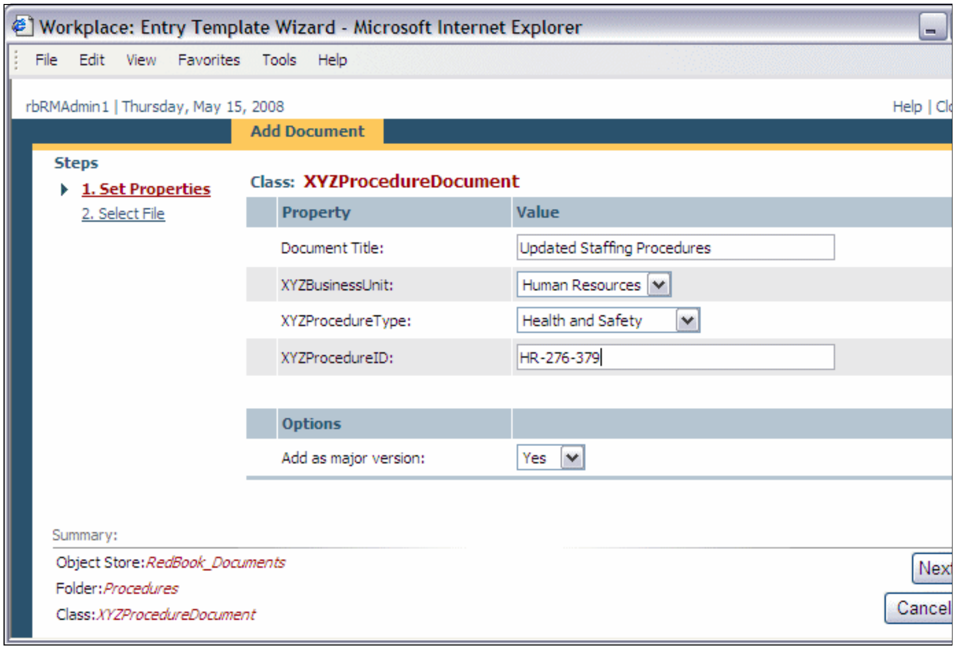
As you can see from the previous list of tasks, the autodeclare event handler requires that the full file plan path and the record class are provided for each document that is to be declared. Because most records management solutions do not expect a user to provide these values for each document being added, the autodeclare event handler as provided by IBM FileNet Records Manager has limited usefulness when the record classification varies from document to document unless this information is provided with each document during ingestion. This approach works for situations where all documents of a given document class in the ROS are classified into the same record category or record folder. However, if you want to dynamically compute the file plan path based on document properties, you need to customize the autodeclare event handler.

Motivation for customizing the autodeclare event handler

As mentioned earlier, in order to implement a more dynamic configuration for automating record declaration, you need to customize the autodeclare event handler. We describe an example that illustrates situations that might require customization.

Computing the file plan path from document properties

In our case study scenario, we have Procedure records that are to be classified in the file plan according to the associated business unit. Depending on the business unit assigned to each document, the document needs to be classified in an appropriate category in the file plan. Figure 5-18 on page 139 shows a Procedure document that has been assigned to the Human Resources business unit by the user who adds the document.


Workplace: Entry Template Wizard - Microsoft Internet Explorer
File
Edit
View
Favorites
Tools
Help
rbRMAdmin1 | Thursday, May 15, 2008
Help | Cl
Add Document
Steps
1. Set Properties
2. Select File
Nex
Cancel

Class: XYZProcedureDocument

Property	Value
Document Title:	Updated Staffing Procedures
XYZBusinessUnit:	Human Resources
XYZProcedureType:	Health and Safety
XYZProcedureID:	HR-276-379

Options	
Add as major version:	Yes

Summary:

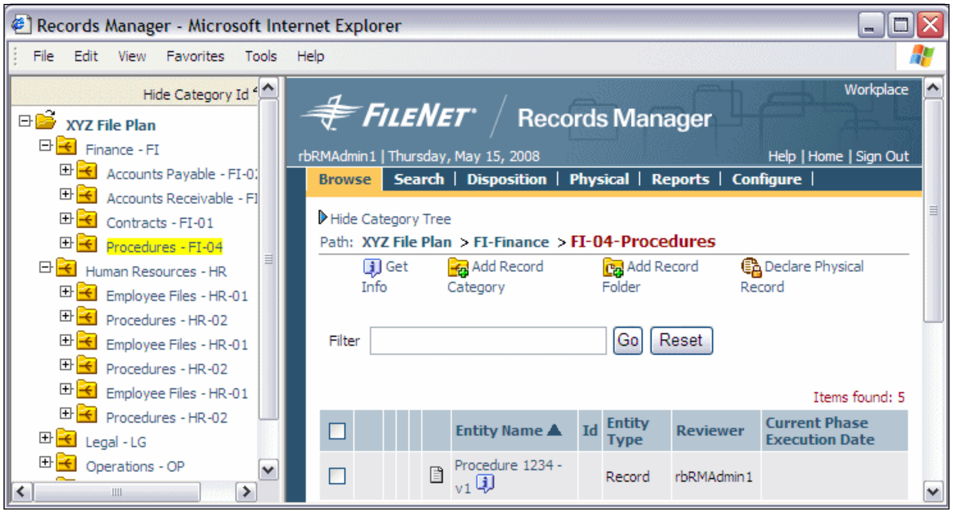
Object Store: *RedBook_Documents*

Folder: *Procedures*

Class: *XYZProcedureDocument*

Figure 5-18 XYZBusinessUnit property can be used to determine classification

The user adding a Procedure document selects the correct business unit from a choice list. In our case study scenario, the business units in the choice list correspond to the level 1 categories in the file plan. Even though we have a single document class in the ROS called XYZProcedureDocument, such documents can be classified into a variety of categories depending on the business unit. Figure 5-19 on page 140 shows how each business unit in the file plan can have a category for Procedures.



XYZ File Plan
- Finance - FI
  - Accounts Payable - FI-0
  - Accounts Receivable - FI
  - Contracts - FI-01
  - Procedures - FI-04
- Human Resources - HR
  - Employee Files - HR-01
  - Procedures - HR-02
  - Employee Files - HR-01
  - Procedures - HR-02
  - Employee Files - HR-01
  - Procedures - HR-02
- Legal - LG
- Operations - OP

Path: XYZ File Plan > FI-Finance > FI-04-Procedures

☐				Entity Name ▲	Id	Entity Type	Reviewer	Current Phase Execution Date
☐				Procedure 1234 - v1		Record	rbRMAdmin1	

Figure 5-19 Separate category for Procedures can exist in each business unit

Similar to how we computed the full file plan path using workflow in a previous example, here we compute the correct file plan path from the value of the XYZBusinessUnit property using Java:

```
FilePlanPath = "/XYZ File Plan/" + XYZBusinessUnit + "/Procedures";
```

In this example, the resulting path from the computation is:

```
/XYZ File Plan/Human Resources/Procedures
```

For more information about adding Java code to the autodeclare event handler, refer to Chapter 9, "IBM FileNet Records Manager Java APIs" on page 213.

Superseding a previous version of a document

To carry this example a bit further, another feature you might want to provide by customizing the autodeclare event handler is to supercede a previously declared version of the Procedure document with the current version that is being checked in. You can implement such a feature by adding to the Java code for the autodeclare event handler.

Here is an example of what this code might accomplish:

1. Find the previous major version of the current document that has just been added.
2. Locate the record object for that version of the document.
3. Supercede that record with the new record that has just been declared.

Additional information on CE event handlers

The autodeclare event handler uses the Bulk Declare Service (BDS), not the IBM FileNet Records Manager API, to declare a record. The BDS provides a faster, more robust mechanism for record declaration and is primarily designed for bulk declaration.

For more details about the BDS, refer to Chapter 9, “IBM FileNet Records Manager Java APIs” on page 213.

5.5 Other mechanisms for ingestion and declaration

In the previous section, we discussed three core mechanisms for implementing record declaration. In this section, we expand our discussion to include other common document ingestion and records declaration mechanisms that provide additional specialized features. Many of these mechanisms and tools can be combined in powerful ways to meet a wide variety of business requirements.

We briefly describe the following mechanisms and how to use them to meet specific document ingestion and record declaration requirements:

- IBM FileNet Capture
- IBM FileNet Email Manager
- IBM FileNet Records Crawler
- IBM Content Collector
- IBM Classification Module
- IBM FileNet Content Federated Services (CFS)
- Custom application using IBM FileNet Records Manager API or BDS

Several of these mechanisms provide both ingestion and native records declaration functionality as well. IBM FileNet Email Manager, IBM FileNet Records Crawler, IBM Content Collector, and IBM FileNet Capture can all be used for ingestion of electronic documents into an IBM FileNet P8 content repository. They also provide configuration options to include record declaration as part of ingestion.

Other mechanisms, such as CFS, only provide for document ingestion into an IBM FileNet P8 content repository and rely on coordinated declaration mechanisms, such as workflow or CE event handlers, to perform the record declaration.

5.5.1 IBM FileNet Capture

When business requirements call for scanning and indexing paper documents, IBM FileNet Capture (Capture) provides a robust mechanism for document entry. Capture can be used to verify scanned images before they are committed to the IBM FileNet P8 content repository, and it has a native indexing component that is well-suited for scanning batches of documents.

You can use Capture as an ingestion mechanism to store the scanned images in an IBM FileNet P8 content repository and then leverage a CE subscription to either launch a workflow or a CE event handler to declare the images as records.

Capture also provides for automatic *record profile* selection based upon document metadata. A given record profile defines the record class and file plan location to be used by Capture during declaration. This method is almost as flexible as workflow or CE event handler, and in most cases, it is faster at content ingestion and record declaration.

The IBM FileNet Capture product also provides support for processing incoming faxes using the IBM FileNet Fax companion product. This product integrates a fax server that receives incoming faxes with the Capture product. These faxes can then be ingested and declared as records using the existing Capture mechanisms.

Business requirements will determine the most appropriate configuration or combination of mechanisms.

5.5.2 IBM FileNet Email Manager

When business requirements call for capturing and declaring e-mail messages from an e-mail system, IBM FileNet Email Manager (Email Manager) can be used to both ingest and declare e-mails as records. Email Manager offers a variety of configuration options that support both server-side rules, client-based rules, and dynamic capture and declaration. The configuration options include pattern matching capabilities and the ability to dynamically construct the full file plan path for record declaration. If appropriate, Email Manager can also be used as an ingestion mechanism that is combined with other automated declaration mechanisms.

With the huge volumes typically seen in an e-mail scenario where Email Manager is used to ingest all e-mails, we do not recommend using events and workflows for e-mail records declaration. However, if you are selectively declaring e-mails, you can combine Email Manager with events and workflow.

Important: During the writing of this book, IBM FileNet Email Manager has been replaced by a new product, IBM Content Collector. Refer to its description in 5.5.4, “IBM Content Collector” on page 143.

5.5.3 IBM FileNet Records Crawler

IBM FileNet Records Crawler (Records Crawler) can be used as an effective mechanism for ingesting and declaring electronic documents from a file system. Records Crawler can be used in a bulk fashion where it is exposed to an existing file system that already contains electronic documents. It can also be used in a more dynamic fashion where the file system is staged with a folder structure that matches the file plan in enough detail to automate the ingestion and declaration of new content that is added by users simply dragging and dropping documents into a set of folders on a network share. Similar to Email Manager, Records Crawler also supports configuration options that include pattern matching capabilities and the ability to dynamically construct the full file plan path.

Important: During the writing of this book, IBM FileNet Records Crawler has been replaced by a new product, IBM Content Collector. Refer to the following description.

5.5.4 IBM Content Collector

Combining IBM FileNet Email Manager and IBM FileNet Records Crawler capabilities, IBM Content Collector (ICC) is an integrated, extensible, modular content collection and archiving solution that enables organizations to take back control and unlock the business value of content, while enforcing compliance and operational policies, and reducing the total cost of ownership.

IBM Content Collector offers two product editions:

- Content Collector for Email: Helps users enhance e-mail value and better manage e-mail growth and risks. It replaces the former IBM FileNet Email Manager product.
- Content Collector for File Systems: Helps users manage the growth and risk of file systems. It replaces the former IBM FileNet Records Crawler product.

IBM Content Collector can be used as an effective mechanism for ingesting and declaring electronic documents from a file system and an e-mail system.

5.5.5 IBM Classification Module

IBM Classification Module (ICM) automates the organization of unstructured content by analyzing the full text of documents and e-mails, and assigning categories to items, such as folders or document classes. These categories can be used by IBM FileNet Records Manager to automatically classify records into the appropriate file plan. ICM combines text analysis classification with rules-based approaches. It can reduce risk by ensuring records management practices are uniformly followed.

5.5.6 Content Federated Services (CFS)

IBM FileNet Content Federated Services (CFS) provides a general mechanism for managing existing content in an electronic repository that is external to the native IBM FileNet P8 Content Engine. CFS-IS is a specific implementation of CFS that works with IBM FileNet Image Services (IS), providing a mechanism to ingest references to the content so that it can be accessed from an IBM FileNet P8 content repository and also can be managed by IBM FileNet Records Manager. CFS-IS can provide an effective configuration for high-volume scanning environments where Capture is used to scan and ingest images into IS at a high rate and several of the images are subsequently federated with IBM FileNet P8 content repository for purposes of managing them as records.

5.5.7 Custom application

For business requirements that go beyond the capabilities provided in the existing mechanisms, a custom application can provide the ultimate flexibility and control for implementing records ingestion and declaration. There are two API sets available for record declaration: the IBM FileNet Records Manager API and the Bulk Declaration Service (BDS).

The IBM FileNet Records Manager API provides objects and methods that implement the core IBM FileNet Records Manager functionality.

The Bulk Declaration Service (BDS) is a separate API that provides functionality specific to the bulk declaration of records and achieves a declaration rate that is significantly higher than the IBM FileNet Records Manager API. The BDS provides enhanced transaction management and handles batches of records more efficiently. It is designed primarily for bulk declaration, and it does not include the complete IBM FileNet Records Manager API functionality. The IBM FileNet Records Manager API is still required for other record-related operations. However, the BDS can be used for any application requiring APIs for record declaration.

For more information about working with custom applications, refer to Chapter 9, “IBM FileNet Records Manager Java APIs” on page 213.

5.5.8 Bulk declaration

Quite often, when initially rolling out a new IBM FileNet Records Manager system, content must be migrated from an existing system into an IBM FileNet P8 content repository, and the content needs to be declared in bulk. Even if the content already exists in the IBM FileNet P8 content repository, as part of implementing IBM FileNet Records Manager, you might need to declare the existing content using bulk declaration. The most direct way to approach this requirement is to write a custom application to perform the record declaration, either alone or in combination with ingesting migrated content. For bulk declaration requirements, BDS is the most appropriate programmatic declaration mechanism.

Bulk declaration can be a one-time requirement that only occurs when initially populating a new system with business records that already exist. Most day-to-day usage scenarios do not require bulk declaration, but they typically rely on dynamic or automated declaration as new content is added one document at a time.

> **Best practice:** When writing a custom application to perform bulk declaration, use the BDS instead of the IBM FileNet Records Manager API to achieve better performance and more robust transaction management during declaration.

5.6 Working with document versions

When a document is declared as a record, it is important to consider the versioning requirements for that document before deciding on a strategy for declaration. Many business scenarios involve documents that are not versioned at all, which makes the issue much easier to manage. But for documents that are versioned, it is helpful to understand how IBM FileNet Records Manager works in regard to multiple versions of a document. As with many aspects of IBM FileNet Records Manager, the business requirements determine how you approach documents with versions.

5.6.1 Declaring versioned IBM FileNet P8 documents

Workplace offers two distinct functions for declaring a record:

- Declare as record
- Declare versions as record

Declare as record works with a single version of a document, declaring the current version of the document as a record.

Declare versions as record allows a user to select from all the existing versions of the current document that have not yet been declared and declare one or more of those versions as a single record.

Depending on how many versions of a document are created, either of these functions can be used to declare records one or more times throughout the life of that document. What is important to understand is that each time that a record is declared, whether it references a single version or multiple versions, these versions are locked down by IBM FileNet Records Manager. Any other versions, major or minor, that have not explicitly been declared, are not locked down and are not under IBM FileNet Records Manager control. Any future versions that are created of the document are not automatically declared as records just because a previous version of the document is declared as a record.

When a record is declared on existing versions, any subsequent versions must be declared as separate records from the original. There is no way to add subsequent versions to a previously declared record. The guiding principle here is that after a record is declared, it cannot be changed. However, a record can be superceded, which is a useful function when dealing with versioned documents in IBM FileNet Records Manager.

5.6.2 Typical versioning scenarios

There are several versioning scenarios when declaring documents as records. These documents might not have been versioned yet, are versioned before being declared as records, or are versioned after they are declared as records.

Documents that are not versioned

One of the common scenarios when dealing with record declaration is to declare the first version of the document and prevent any additional versions from being created. In this case, there is no need to be concerned with managing multiple versions. This situation is a typical scenario for scanned images where you do not expect users to perform a checkout and checkin to create a new version. This scenario also applies to documents that are not added to the IBM FileNet P8 content repository until the final version is produced or for e-mail messages, which are never modified after they have been sent or received.

Example: Inbound customer correspondence

You want to declare as records all inbound customer correspondence related to an insurance claim. Customer correspondence arrives in either paper or e-mail form. Each paper letter is scanned and declared as a record as soon as it is stored in the content repository. Likewise, each piece of e-mail is captured and declared as a record. None of these documents will ever be versioned.

Documents that are versioned before declaration

A typical authoring scenario might involve adding a document to the IBM FileNet P8 content repository and having the document authors work on the document before declaring a designated final version as a record. In this scenario, you allow designated authors to check out and check in the document as many times as needed to produce a final version. After the final version is checked in, only that version is declared as a record. It is up to the authors, or a process that you define, to clean up any of the draft versions.

Example: Authoring a new contract

You want to author a new contract that will be sent to a customer; only the final version of the contract, the one that is actually sent to the customer, is required to be declared as a record. The authoring process requires multiple revisions by several authors. The authors check out and check in the contract document many times before the final version is produced. When the primary author checks the final version into the system, the author declares the current (final) version of the contract document as a record and optionally deletes all previous versions. After the final version is declared as a record, the authors can no longer check out and modify the contract document.

A variation of this scenario is to wait until the final version is checked in but to declare all major versions as a single record. There are an unlimited number of possibilities depending on your business requirements and your authoring process. Remember that any versions not declared as records are not under IBM FileNet Records Manager control and must be managed as appropriate per the business requirements. For example, if you declare all major versions as a record, the minor versions are not automatically deleted. You have to decide what you want to do with the minor versions.

Documents that are versioned after declaration

Another scenario might involve a document authoring process where newer versions are created after a specific version is declared as a single record. This example is a common scenario where a single document might be updated periodically to produce an up-to-date version of a particular document. But each time that the document is updated, that version needs to be declared as a record. In this scenario, as soon as the next major or released version has been checked in, the specific version is declared as a record. It is up to the business requirements to determine what happens with the previously declared versions of the same document. From the IBM FileNet Records Manager perspective, each version that is declared as a record separately is considered a separate record. The original record from a previous version is never modified.

Example: Procedure documents are updated annually

You have a procedure document for the Human Resources department that must be updated once a year and declared as a record each time. Each year, when the revisions are completed, one of the authors declares the latest version as a record. Each version that has been declared as a record is maintained in the file plan as a separate record.

For this scenario, a common requirement is to supercede the record of the previously declared version. However, the system does not do this step automatically. A user can manually supercede an existing record, or you can write custom code to automate this requirement. For a discussion of using the autodeclare event handler to supercede a record, refer to 5.4.4, “CE event handlers” on page 136.

5.7 Performance considerations

The speed at which the system can ingest and declare records is an important consideration when planning the overall design and size of your system. When sizing your system, consider both average and peak ingestion rates, as well as declaration rates, and make sure that your system is sized appropriately to handle the anticipated inbound volume.

Ingestion performance and declaration performance usually become topics for concern when dealing with bulk declaration or with ingestion mechanisms, such as Email Manager, that can potentially generate a continuous stream of large volumes of data. To achieve the best performance for bulk declaration scenarios, use the BDS instead of the IBM FileNet Records Manager API to declare records from a custom application. Refer to 9.6, “Bulk Declaration Service (BDS)” on page 232 for details.

Most day-to-day usage scenarios involve dynamic or automated declaration of one document at a time with user interaction for each document. In these cases, the speed with which the declare operation completes is not a rate-limiting factor.

6

Records disposition

Timely records disposal is a critical facet of records management that helps companies and organizations achieve compliance with internal, industry, and governmental regulations and laws. Timely disposal also reduces costs in areas, such as litigation, operations, and records storage. In this chapter, we begin with a focus on the importance of records disposition, and then we describe in detail the key features and components of IBM FileNet Records Manager that are related to disposition.

In this chapter, we discuss the following topics:

- Records disposition
- Disposition schedules
- Disposition Sweep
- Initiating and completing disposition actions
- Automatic destruction (Auto Destroy feature)

6.1 Records disposition

Records disposition defines what happens to records at the end of their retention period. As already discussed in Chapter 3, “Retention schedules and file plans” on page 53, retention rules and policies describe how long records must be retained before they are disposed. Retention periods normally are directed by laws, regulations, or business procedures and policies. *Disposition* refers to the processes or actions taken on a record or group of records at the end of a retention period.

Even though destruction is the ultimate goal in many record disposition processes, it is only one of several options available. The nature of an organization, the laws, regulations, and policies dictate what an organization is required to do with its records, including the type of action that needs to be taken for successful records disposition. The type of record also plays a key role in determining what to do with a record. Records disposition is not only about timely destruction, but it can also be about the preservation and archiving of certain records. For example, a government agency might be required to transfer historical records to the National Archives of United States as the final step in the disposition process. Preserving records during the transfer process differs dramatically from preparing to permanently destroy records.

Note: The term *disposal* is often equated with destruction. However, in the context of records management, *disposal* can refer to any disposition process, which can include destruction, transfer for purposes of preservation, or review of permanent records.

6.1.1 Importance of records disposition

In the past, many organizations have operated under the assumption that they must keep records indefinitely. Without reasons, records were often kept just in case they were ever needed at an indeterminate time in the future. For a variety of reasons, developing and enforcing consistent records management policies, including proper disposal of records at the end of their useful life, was not viewed by companies as a high priority.

This approach (or lack of approach) to managing records is no longer acceptable. Not only is the amount of information that organizations have to manage exploding, but more importantly, the legal liability and cost associated with retaining this information can be extraordinary.

An organization that does not have or does not enforce its records management policies is exposing itself to significant legal liabilities. These liabilities include:

- Spoliation charges
- Discovery liabilities
- Discovery costs
- Fines and penalties

Savvy plaintiff legal counsel knowingly use high discovery costs as a settlement tactic. They know that the cost and resources associated with looking through a mountain of information is prohibitive. It does not matter if there is nothing relevant in that mountain, you still have to look through it. Counsel in litigation also know that if given the chance to look at everything (unrestricted access to all the other party's records), they have a high probability of finding damaging information that will help their case.

Table 6-1 highlights the key legal and cost factors associated with records management disposal policies that are nonexistent or not enforced.

Table 6-1 Factors related to poor records management practices

Behavior or situation	Legal factors	Cost factors
Keeping records too long or not destroying them at all	► Exposure to legal liability ► Target for legal discovery	► Costly to discover ► High storage costs
Destroying records too soon	► Potential for spoliation charges ► Inability to produce	► Costly to reproduce
Lost records	► Potential for spoliation charges ► Charges of ineffective records management ► Inability to produce	► Costly to reproduce

If there is not a clear legal, regulatory, or business justification for retaining information, you must dispose of the information.

6.2 Disposition schedules

In this section, we provide an overview of the various elements and components related to records disposition in IBM FileNet Records Manager and how these elements and components relate to the lifecycle of a record.

The following list identifies the main activities to perform in order to successfully implement disposition in the IBM FileNet Records Manager environment:

1. Configure disposition schedules to represent the relevant retention rules in the organization’s retention schedule.
2. Apply the disposition schedules to the file plan to affect the records to be disposed.
3. Run Disposition Sweep.
4. Initiate disposition on the desired entities in the file plan.
5. Complete the disposition process.

When a record is first declared, the major focus from a business perspective is to ensure that the record is protected from accidental or unauthorized destruction and to ensure that the record is readily accessible by authorized users. Records in this early stage are often referred to as *active records*. Depending on the nature of the records and the retention rules applied to them, at a certain point, records are considered inactive, meaning that they are no longer needed for the regular, current business activities of the organization. It is usually this transition from active to inactive that is related to the disposition process. The retention rules of an organization typically define retention periods for the various types of records that an organization keeps, indicating the length of time these records must be kept after they are no longer active.

Depending on the nature of the business requirements, a disposition schedule can define a simple, single-phase disposition process, or it can define an elaborate, complex, multi-phase disposition process. In order to understand the options available, we look at the components of a disposition schedule in more detail.

6.2.1 Disposition schedule

IBM FileNet Records Manager uses disposition schedules in order to control the retention and disposal of records. Disposition schedules encapsulate the retention rules for records and instructions for the disposal of records at the end of the retention period. The primary components of an IBM FileNet Records Manager disposition schedule are:

- Disposal trigger that specifies a trigger condition and aggregation level
- Cutoff as defined by an offset value and a cutoff base
- One or more disposition phases, each of which defines a retention period
- Actions associated with each disposition phase

Figure 6-1 illustrates the basic relationship of these aspects of a disposition schedule in the context of a record's lifecycle. After a record is declared, it is under the control of the configuration specified in the file plan. However, disposition for the record does not engage until a trigger condition has been met. The disposal trigger signals that the remaining disposition parameters can be calculated based on the configuration of the disposition schedule. Depending on how the disposition schedule is configured, there might be an offset before cutoff is achieved. *Cutoff* is a demarcation that the retention period as specified in the phases of disposition can begin. After the retention period for a disposition phase has elapsed, the entity being disposed is ready for the disposition action associated with that phase. The disposition action for a phase can then be initiated and completed.

We discuss each of these topics in more detail next.

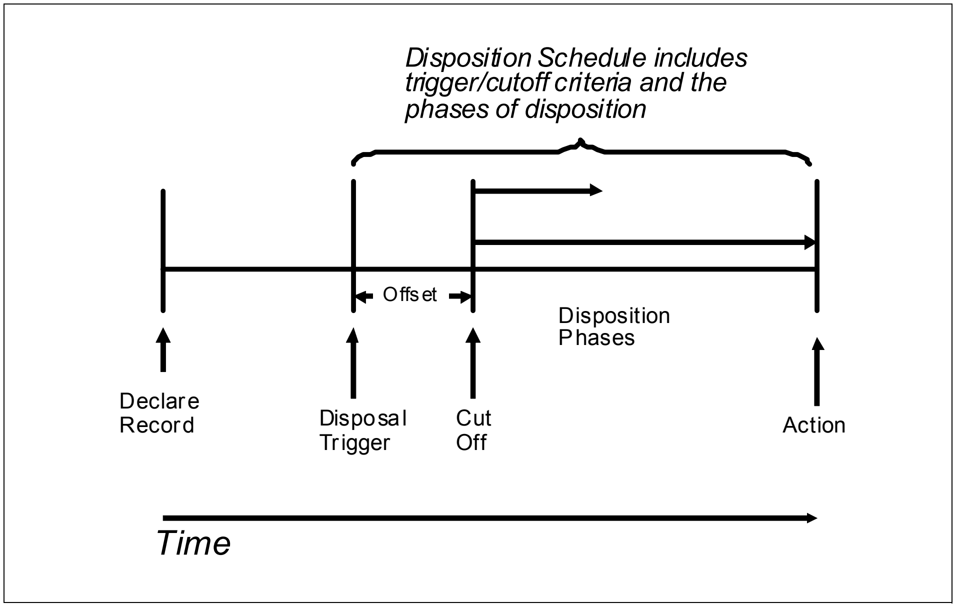

Disposition Schedule includes trigger/cutoff criteria and the phases of disposition
Offset
Disposition Phases
Declare Record
Disposal Trigger
Cut Off
Action
Time

Figure 6-1 The relationship of a disposition schedule to the lifecycle of a record

The disposition schedule brings together several configuration elements and offers a wide range of options in terms of how disposition is defined for a given schedule. In addition, the disposition schedule defines the disposition criteria, but the system relies on *Disposition Sweep* to compute essential disposition parameters that can only be calculated after the trigger condition is met. We discuss the role and use of Disposition Sweep in a later section.

Before we discuss the various configuration elements and how they come together in a disposition schedule, it might be instructive to look at a couple of fundamental properties of a disposition schedule.

Disposition schedule name

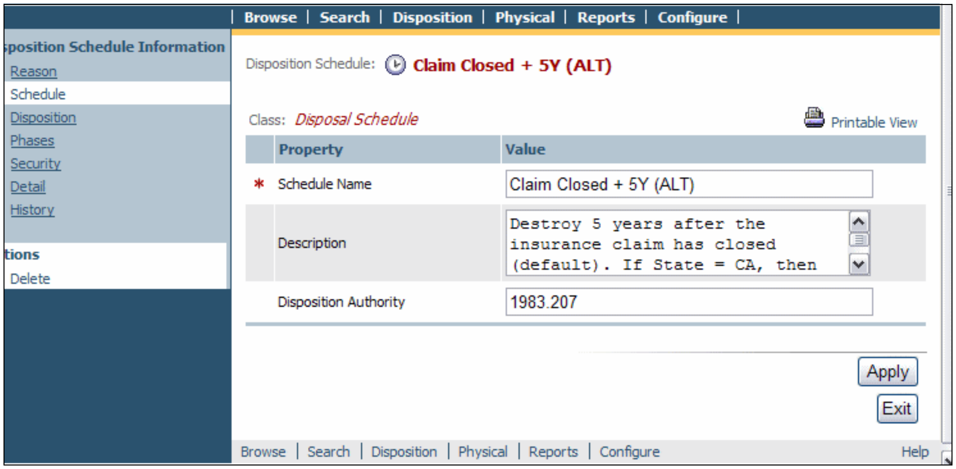
Each disposition schedule must have a unique name. It is useful to provide a name for the disposition schedule that describes the essential behavior of the retention rule being implemented. How you name your disposition schedules will be determined by the overall design of your file plan and the various retention rules that you intend to implement. The disposition schedule name is used to identify the appropriate disposition schedule when assigning it to the file plan.

> **Best practice:** Determine a consistent naming convention for your disposition schedules based on the nature of your retention rules and how you have organized them in your overall retention schedule.

Disposition authority

The disposition authority provides a reference or citation to why this disposition schedule has been adopted. The citation can be an external reference to a specific law or industry regulation, it can be strictly internal and represent an organization's business policies or practices, or it can be a list of several reference sources. The disposition authority property is solely for purposes of documentation. In IBM FileNet Records Manager, the disposition authority is a string property that is used only for reference purposes and does not affect the behavior of the disposition schedule.

Figure 6-2 shows these key properties that identify a disposition schedule. The schedule name is short and descriptive and, in this particular example, includes references to the trigger and the retention period. The description property is used to describe the retention rule in more detail. The disposition authority property, in the example shown here, is a numerical reference to an internal policy. The three properties shown here can be used together to identify and document the purpose of the disposition schedule.


Browse | Search | Disposition | Physical | Reports | Configure
sposition Schedule Information
Reason
Schedule
Disposition
Phases
Security
Detail
History
tions
Delete
Printable View
Apply
Exit
Browse | Search | Disposition | Physical | Reports | Configure
Help

Disposition Schedule: Claim Closed + 5Y (ALT)

Class: *Disposal Schedule*

	Property	Value
*	Schedule Name	Claim Closed + 5Y (ALT)
	Description	Destroy 5 years after the insurance claim has closed (default). If State = CA, then
	Disposition Authority	1983.207

Figure 6-2 Properties of a disposition schedule used for identification

Next, we discuss the various configuration elements that define the behavior of the disposition schedule.

6.2.2 Disposal triggers

A *disposal trigger* signals that record disposition can begin. Often, the trigger condition itself is a date value that is also used for calculating cutoff and retention. However, the trigger condition is fundamentally a signal that a specific condition has occurred that will allow the remaining disposition parameters to be calculated. For example, a contract document might be triggered for disposition by the contract expiration date being set. While a contract is still active and not yet expired, the date value remains null. As soon as the contract expiration date is set, this action is the trigger for disposition.

In addition to a trigger condition, a disposal trigger determines the aggregation level for purposes of disposition. The aggregation level indicates which entity type is being disposed (an individual record, a volume, a record folder, or an entire record category).

In IBM FileNet Records Manager, a disposal trigger can be defined as one of four types:

- Internal event trigger: These triggers are the most commonly used triggers. The triggers are tied to the metadata of the entities being disposed. After the trigger condition is satisfied, Disposition Sweep can calculate the remaining disposition parameters that determine cutoff and the retention period. Internal event triggers are often based on a date property on the entity being disposed. The scenarios for the case study used in this book rely on internal event triggers.

Examples: Contract expiration date, date closed, and date superceded

- External event trigger: External events are one-time events that occur outside the system but that can directly impact the cutoff and disposition of entities. The time of the event is not known at the time that the disposition schedule is created, but it is entered later by an authorized user when the event occurs. As soon as the external event occurrence date is set, Disposition Sweep can calculate the remaining disposition parameters that determine cutoff and the retention period.

  Examples: Life of company and life of a plan

- Recurring event trigger: *Recurring events* are events that recur automatically after a predefined time interval. They are typically associated with vital records and are used to trigger periodic reviews of these records.

  Example: Annual reviews

- Predefined: Predefined events are similar to external events, except in this case, the date is known ahead of time.

  Example: June 30, 2011

Aggregation

One important aspect of a trigger is the *aggregation* to which the trigger applies. This aggregation is especially important when defining an internal event trigger, because the properties available for the trigger condition depend on the aggregation selected. The following aggregation levels are available for internal event triggers:

- Record category
- Record folder
- Volume
- Record

Aggregation has a direct impact on the behavior of Disposition Sweep. The aggregation selected for the trigger determines on which entities Disposition Sweep operates. It also determines at which level entities are batched for disposition processing.

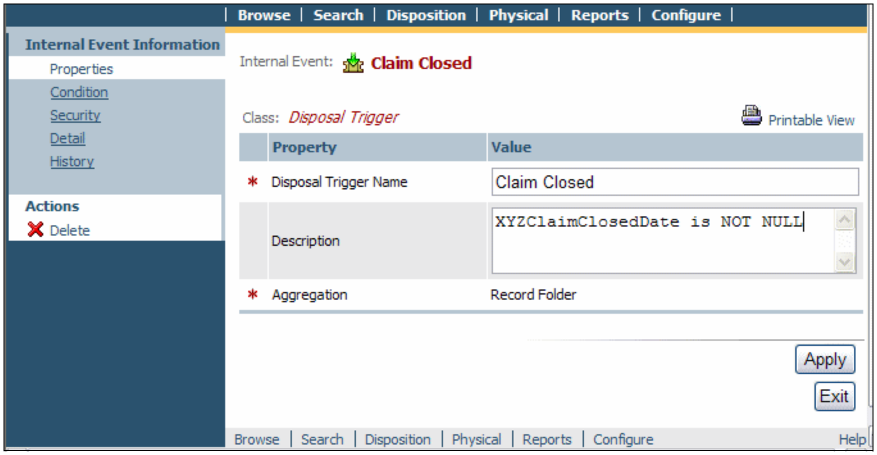
Example: Internal event trigger

To illustrate the concept of a trigger and aggregation, we use the insurance claim file scenario from our case study. If you recall from previous chapters, our case study is designed so that insurance documents are disposed based on the claim closed date. As part of our file plan design, we group claim documents in separate records folders that identify each claim. When it is time to dispose of the claim documents for a given claim, we dispose of the claim folder and its entire contents (all of the documents that the claim folder contains). Because of this design choice, we can set up an internal event trigger that specifies folder-level aggregation to trigger on the claim closed date. This design choice relies on the claim closed date property being stored on the claim folder. Hence, the metadata used for triggering and calculating disposition is internally stored as part of the entity being disposed.

The internal event trigger has three key properties:

- Disposal trigger name
- Aggregation
- Condition

The *disposal trigger name* is used to identify the trigger when assigning it to a disposition schedule. The aggregation determines the entity type upon which the trigger condition is applied and also determines which properties are available for the trigger condition. The condition defines the condition for the trigger. Figure 6-3 shows a screen capture of the internal event trigger properties for our claim closed example, in which the aggregation is record folder.


Browse | Search | Disposition | Physical | Reports | Configure
Internal Event Information
Properties
Condition
Security
Detail
History
Actions
Delete
Printable View
Apply
Exit
Browse | Search | Disposition | Physical | Reports | Configure
Help

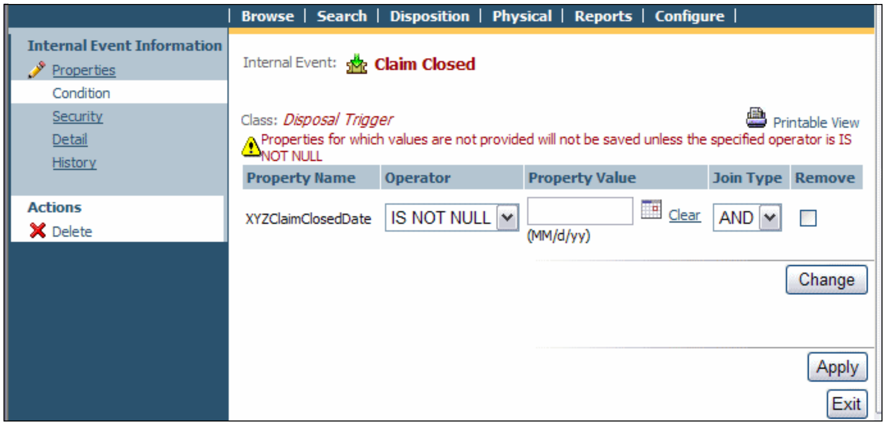
Internal Event: Claim Closed

Class: *Disposal Trigger*

	Property	Value
*	Disposal Trigger Name	Claim Closed
	Description	XYZClaimClosedDate is NOT NULL
*	Aggregation	Record Folder

Figure 6-3 Properties for an internal event trigger

Figure 6-4 shows a screen capture of the condition for this trigger. In this case, XYZClaimClosedDate is a property of the claim folder. We can select this property for the trigger condition, because we chose record folder for aggregation.



Internal Event: Claim Closed

Class: *Disposal Trigger*

Properties for which values are not provided will not be saved unless the specified operator is IS NOT NULL

Property Name	Operator	Property Value	Join Type	Remove
XYZClaimClosedDate	IS NOT NULL	(MM/d/yy)	AND	□

Figure 6-4 Condition for an internal event trigger

6.2.3 Cutoff

Cutoff is a condition that marks the beginning of the phases of disposition. It is a signal to the system that the entity being disposed is no longer active and can transition to the disposition phases. The concept of cutoff is typically more relevant in the realm of physical records, where, for example, at a certain point in time, a box containing files might be cut off from users adding any more records to that box. In the realm of electronic records, the concept is still relevant, and it must be taken into account when configuring a disposition schedule.

One way to think of cutoff is that the period before cutoff represents phase zero (the phase before phase one) of disposition. When an entity reaches cutoff, Disposition Sweep not only sets the cutoff date, but it also computes other disposition properties for the entity indicating that cutoff has been reached and that the entity has transitioned to the first disposition phase. When cutoff is achieved, there is no going back - the cutoff date will not be recalculated. If the entity is a container, when cutoff is achieved, the entity will also be closed. Typically, you do not add more records to a container that is closed.

Computing cutoff

Cutoff includes a date property that is computed by Disposition Sweep. It is calculated based on cutoff base and offset.

Cutoff formula: Cutoff date = cutoff base + offset

When configuring the cutoff for a disposition schedule, you have the option to have the cutoff date calculated based on a specific date property that you select, which serves as the cutoff base, and the offset, which is an interval added to the cutoff base.

Cutoff base

Unless you select a specific date property for the cutoff base, the default *cutoff base* is configured for Event Date, which is the time and date that Disposition Sweep detected the event and performed the disposition calculations for the entity. In most cases, you want to select a *specific date property* to serve as the cutoff base. For example, in our case study scenario for contracts, the contract expiration date is selected as the cutoff base, because we want to control disposition based on this property. This approach removes any dependency on when we run Disposition Sweep. Whenever we decide to run the Disposition Sweep, it accurately computes the appropriate cutoff date.

Offset

The *offset* is a time interval that gets added to the cutoff base when Disposition Sweep computes the cutoff date. The offset value is used to delay cutoff of an entity. In certain configurations, the offset value itself can serve as the retention period when combined with a zero interval for the phases of disposition.

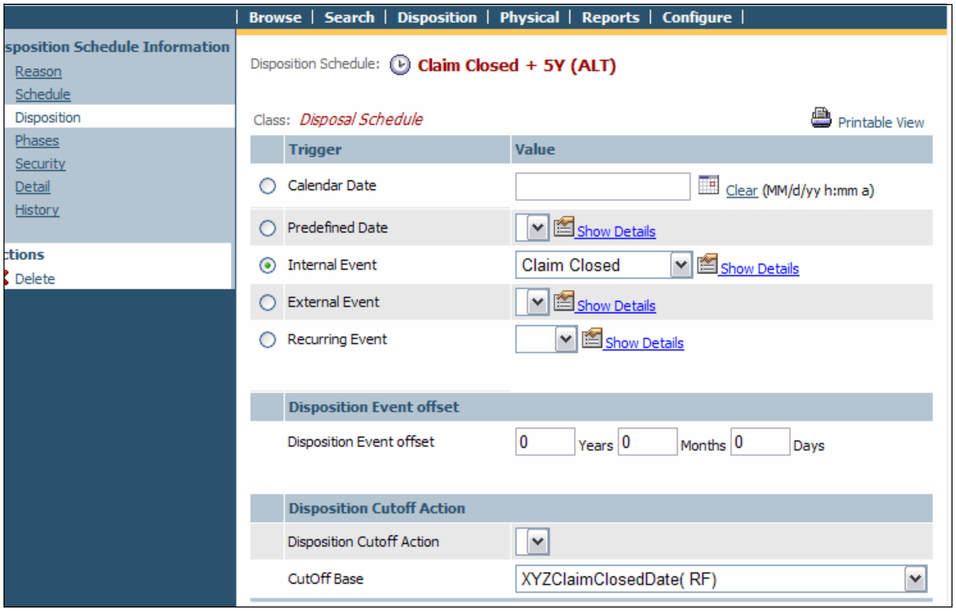
Cutoff action

The *cutoff action* is a specific disposition workflow that is automatically launched when Disposition Sweep is run and the entity in question is ready for cutoff. The intention of the cutoff workflow is to allow for a review before cutoff is finalized, providing an opportunity for a user to manually adjust the actual cutoff date. The cutoff action is an optional feature and can be useful, depending on your business requirements for disposition. In most disposition scenarios, the cutoff action is not required or desired (for example, you might not want your users to be able to manually adjust the cutoff date). The *Cutoff Workflow* is provided with the product as a sample process that is associated with the cutoff action.

If there is no cutoff action specified for a disposition schedule, Disposition Sweep automatically computes the cutoff date. After the cutoff date occurs, Disposition Sweep also computes other phase information properties for the entity being disposed that indicate cutoff has occurred.

Configuring cutoff in the disposition schedule

Cutoff is configured as part of the disposition schedule. Figure 6-5 shows the selection of the internal event for the disposal trigger in the example scenario of our case study, specifically, the Claim Closed trigger. The offset interval is set to zero, and there is no cutoff action specified. The cutoff base is set to the selected date property, XYZClaimClosedDate, which happens to be a property associated with a record folder (RF).



Disposition Schedule: Claim Closed + 5Y (ALT)

Class: *Disposal Schedule*

	Trigger	Value
○	Calendar Date	(MM/d/yy h:mm a)
○	Predefined Date	
●	Internal Event	Claim Closed
○	External Event	
○	Recurring Event	

Disposition Event offset

Disposition Event offset: 0 Years 0 Months 0 Days

Disposition Cutoff Action

Disposition Cutoff Action:

CutOff Base: XYZClaimClosedDate(RF)

Figure 6-5 Disposition schedule: Using internal event trigger, offset, and cutoff base

In this example, because the offset is zero and there is no cutoff action, when Disposition Sweep runs and the trigger condition is met, cutoff is achieved immediately.

6.2.4 Disposition phases and actions

In IBM FileNet Records Manager, the process for disposing of records is encapsulated in the phases of the disposition schedule. A disposition schedule contains one or more disposition phases. Associated with each phase of disposition is a default retention period as well as an action. When the retention period for a particular phase has elapsed, the specified action is ready to be performed. Phases of a disposition schedule are sequential. Figure 6-6 shows multiple phases for a disposition schedule and how the default retention periods for the phases relate to each other. The default retention period for each phase is relative to cutoff. This particular example shows three disposition phases: the first phase specifies a review, the second phase specifies export, and the final phase specifies destroy.

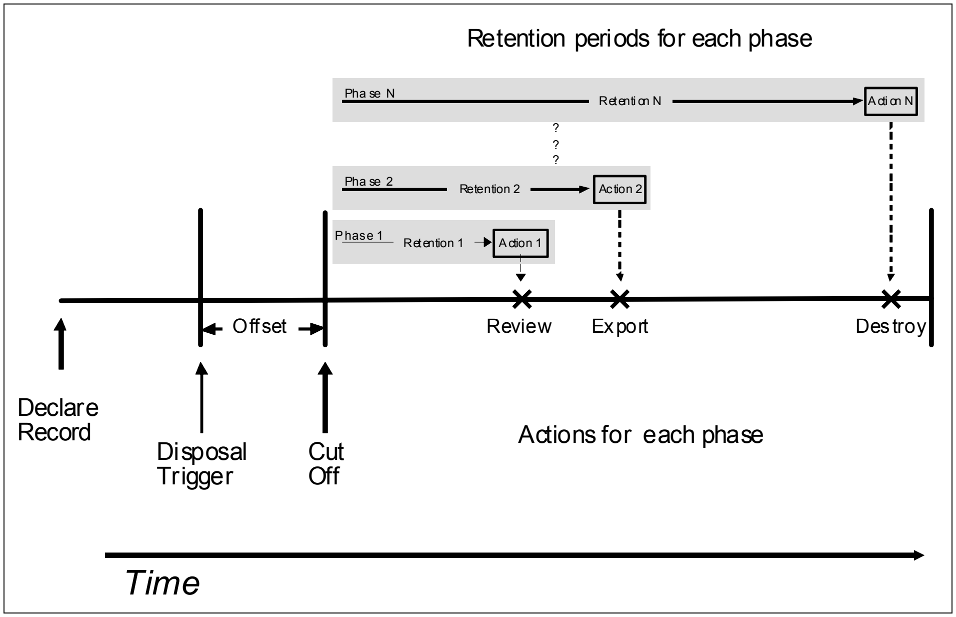


Figure 6-6 Retention periods and actions for each phase of disposition

While IBM FileNet Records Manager allows you to configure a disposition schedule with multiple phases, the most common use cases call for only one disposition phase that completes the destruction process. The number and type of phases are driven by your business requirements and your underlying records management policies and procedures.

Disposition actions

The *disposition action* defines what needs to happen to the entity being disposed when the retention period for a given disposition phase is elapsed. In most cases, the disposition action is implemented by a workflow. Because compliance solutions can significantly benefit from a process-driven approach, IBM FileNet Records Manager is tightly integrated with the underlying BPM capabilities of the IBM FileNet P8 Platform.

The following action types are used for disposition actions that are associated with the phases of disposition:

- Review
- Export
- Interim Transfer
- Transfer
- Destroy
- Auto Destroy

We describe each of these action types in more detail and include information about the predefined workflows that are associated with each action type.

Review

The *Review* action indicates the need for a general disposition review of records by records management staff or by an appropriate designated reviewer who is associated with the records. During the review process, the reviewer might place records on hold or update the record properties to affect the disposition status of the records. A review action alone typically does not result in any of the other outcomes associated with any of the following actions, but it simply provides an opportunity for review. Often, the other actions include their own review or approval step as part of the action for that disposition phase, making a separate review phase and action unnecessary. The *Disposition Review Workflow* is provided with the product as a sample disposition review process that is associated with this action.

Export

The *Export* action indicates that records must be exported or copied to another system or repository. The export process involves creating a copy of the records and optionally the metadata (in the case of electronic records, you can export the content and the metadata; in the case of physical records, you can only export the metadata) that can later be imported into an external system or repository. Export only involves making a copy; it does not remove or destroy the records. The *Export Workflow* is provided as a sample export process that is associated with this action.

Interim Transfer

The *Interim Transfer* action indicates that records are to be transferred to another location while maintaining the record information (metadata only) in the current system.

For electronic records, interim transfer includes these steps:

1. Make a copy of the electronic record and its properties for export.
2. Make the exported content available for import to a storage repository outside the IBM FileNet P8 system.
3. Delete the electronic content from the IBM FileNet content repository while maintaining the record information.
4. Update the location information for the record.

The interim transfer of electronic records results in the content not being directly retrievable through either IBM FileNet Records Manager or Workplace, because the content is deleted as part of completing the transfer. However, the record information remains intact.

For physical records, the process includes a manual step to ensure that the physical transfer has been completed before the record properties are updated with information about the new location. For example, interim transfer might involve moving a set of boxes from a local storage area to a remote warehouse. For physical records, there is no electronic content to export. The *Interim Transfer Workflow* is provided as a sample interim transfer process that is associated with this action.

Transfer

The *Transfer* action indicates that records must be transferred to an external system or repository. The transfer process usually involves first making a copy of the records and ensuring that they safely arrive at their intended external destination. After the records have been copied to the external system, they are then deleted from the current system. With electronic records, the process includes transferring both content and metadata. With physical records, the transfer process includes both the physical objects and the associated metadata. The transfer action is typically used when regulations require that records are transferred to an archival institution for permanent preservation. The *Two Step Transfer Workflow* is provided as a sample transfer process that includes the export followed by destruction that is associated with this action.

Destroy

The *Destroy* action indicates that the records must be destroyed from the system according to a defined destruction process. The destroy process often includes a review step as part of the destruction process to allow a reviewer to approve or reject the records before the records are actually destroyed. Records that are rejected can be placed on hold or updated to prevent their destruction if appropriate. After records are approved for destruction, electronic records are automatically destroyed. Physical records require a manual destruction step where the records staff must confirm the physical destruction before the metadata is destroyed. In the case where the system is configured to retain metadata upon destruction, only a logical delete of the metadata is performed after the content has been destroyed. The *Destroy Workflow* is provided as a sample destruction process that is associated with this action.

Note: The *retain metadata* option can be enabled for a file plan so that whenever an entity (record or container) is deleted or destroyed, the metadata is retained after the content is destroyed. With this option, the metadata is only logically marked as deleted, but it is actually retained in the underlying database. Without this option, all metadata that is associated with the entity being destroyed is permanently deleted. When browsing the file plan, logically deleted records are not shown. However, these records can be shown by performing a search with the criteria IsDeleted = true.

Auto Destroy

The *Auto Destroy* action indicates that the records must be destroyed from the system immediately without relying on a destruction process or user intervention. This action is implemented by an option included in Disposition Sweep and is typically used for certain types of electronic records that do not require manual approval before destruction. For example, in a scenario where e-mail is archived in the records management system and has a fixed retention period, it can be a business requirement to immediately destroy the records after the retention period has elapsed, as long as the records are not placed on hold. Disposition Sweep includes a function that initiates and completes this action automatically. There is no workflow associated with this action. We discuss the Auto Destroy feature in more detail in 6.5, “Automatic destruction (Auto Destroy feature)” on page 181.

> **Note:** Auto Destroy is a feature that is only available for IBM FileNet Records Manager 4.5 or later versions.

Ordering of multiple disposition phases

The IBM FileNet Records Manager system uses this action type to ensure that phases are configured in a logical order when adding multiple phases to a disposition schedule. For example, you do not want a review phase or an export phase to come after a destroy phase, because there are no records to review or export if you destroy the records first. When adding phases to a disposition schedule, the system ensures that phases are configured in an appropriate order by checking the phase actions for consistency. The major constraint that is applied is that you cannot have any disposition phases after a phase that includes destruction or transfer.

Screening

Screening is an optional process available for any disposition phase that launches a workflow to allow a reviewer to prescreen the entities being disposed before actually initiating the disposition action for that phase. Screening is useful for scenarios where the records management staff is required to screen all records before initiating a formal departmental review or a more complex destruction process. The *Screening Workflow* is provided as a sample screening process. Screening is an option that is enabled separately for each disposition phase and is not a phase itself. This option is typically not used unless the business requirements call for it.

Note: The screening option allows for only a single workflow to serve as the screening workflow system-wide. This option is primarily designed for a single, uniform screening process that is applied across the entire system on those disposition phases where it is enabled. Although you can customize the screening workflow, you cannot apply different screening workflows to different disposition schedules or phases. You can only configure a single screening workflow *system-wide*.

6.2.5 Disposition workflows

The disposition actions listed in the previous section (except for Auto Destroy) are associated with workflows that provide the process-driven behavior required to successfully complete the action on the entities to be disposed. Because IBM FileNet Records Manager is fully integrated with the underlying business process management capabilities of the IBM FileNet P8 Platform, you can customize these workflows to meet specific business requirements associated with the various disposition actions.

Here, we list the workflows that are provided with IBM FileNet Records Manager. These workflows are briefly described in the previous sections when discussed in the context of their corresponding actions or processes.

Phase action workflows

Phase actions include the following workflows:

- Disposition Review: Associated with the Review action
- Export: Associated with the Export action
- Interim Transfer: Associated with the Interim Transfer action
- Two Step Transfer: Associated with the Transfer action
- Destroy: Associated with the Destroy action

Other disposition workflows

Other disposition workflows include:

- Cutoff: Associated with the cutoff action
- Screening: Associated with the screening option

6.2.6 Alternate retention

Alternate retention is a feature that allows you to add flexibility to your disposition schedules in situations where certain entities that are being disposed have similar disposition requirements, but their retention periods differ based on specific conditions associated with each entity. For example, a national insurance company might have to follow federal regulations and laws concerning the disposition of insurance claims that apply to all states. However, specific states might have statutes that supercede federal law. Rather than having a single disposition schedule that applies to most states and separate disposition schedules for each state that has different regulations, a single disposition schedule can be created that takes advantage of the alternate retention option within IBM FileNet Records Manager.

Every disposition phase has a default retention period, which is used to calculate retention based on cutoff. Unless otherwise configured with alternate retention, the disposition schedule always uses this default retention parameter.

Alternate retention provides a means to apply conditional retention to entities that are governed by the same disposition schedule, overriding the default retention specified for a phase when certain conditions are met. Alternate retention provides two benefits:

- Multiple conditions specifying alternate retention periods based on conditions related to the metadata of the entities being disposed
- The ability to select the cutoff base for the retention period being applied to the disposition phase instead of relying on the cutoff date for computing the default retention

One of the scenarios in our case study uses alternate retention to illustrate an example where you have insurance claims that require different retention periods depending upon the state for each claim. In this example, most states require that auto claims are retained for a period of 5 years (default retention). However, two states, California and Florida, have different requirements for the retention period. Rather than defining three disposition schedules, one for California, one for Florida, and one for the rest of the states (or more likely 50 different schedules, one for each state), alternate retention allows you to define one schedule and build in the conditional business logic to handle different state requirements. Figure 6-7 illustrates how to use alternate retention in the case study file plan. The same disposition schedule is inherited by each of the claim folders, but depending on the value for the state property on each folder, either the default retention or one of the alternate retentions applies.

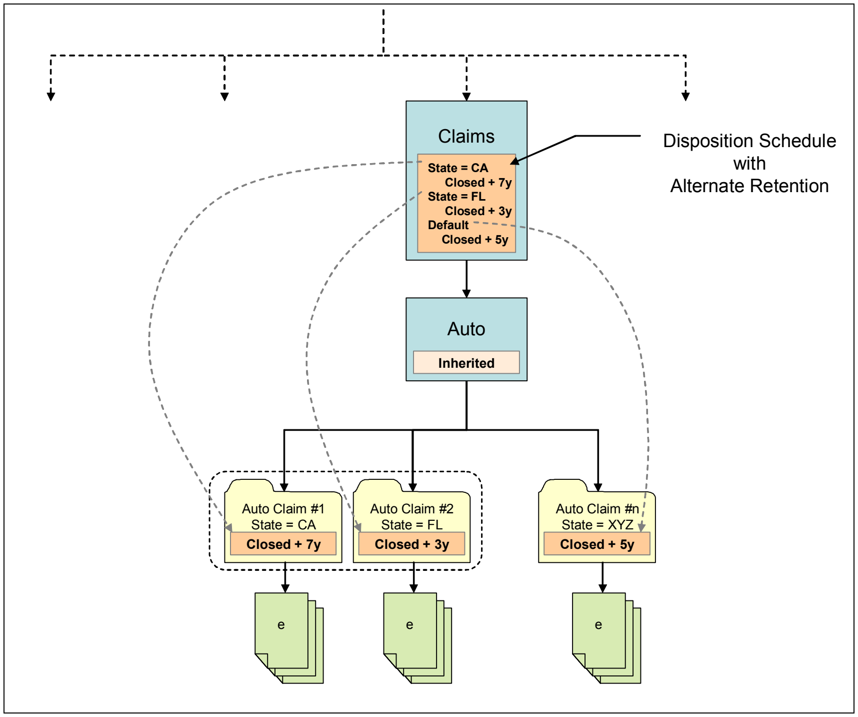


Figure 6-7 Alternate retention example

Without alternate retention, we have to add a level in the file plan that has a category for each state, and we have to apply different disposition schedules to different states. Alternate retention eliminates the need for this extra level of containment and solves the business requirement in a more elegant and efficient manner.

6.2.7 Assigning disposition schedules to the file plan

File plans by their very nature are hierarchical. They are made up of a combination of record containers and the records themselves. Their design reflects the business classification schema that an organization uses to manage its records. Top-level nodes are typically coarse groupings of related entities, and subsequent levels of the file plan further refine the groupings, which are typically based on how the various types of records in the organization are related to retention requirements.

As discussed in Chapter 3, “Retention schedules and file plans” on page 53, disposition schedules are typically assigned to the categories in the file plan and are inherited by lower levels where they eventually apply to the entities being disposed.

Understanding the role of inheritance and aggregation

In a typical file plan, records (or other entities being disposed, such as records folders) do not have a disposition schedule assigned directly to them. Instead, they inherit their disposition behavior from the closest parent container that has a schedule assigned. You associate the disposition schedules with the appropriate categories in the file plan and the entities beneath that category are governed by that disposition schedule. If there are specialized subcategories or folders under a parent container that require different disposition schedules, you can associate schedules with containers at the lower levels in the file plan that override the schedules above it in the hierarchy. In other words, the inheritance of disposition schedules can be overridden at lower levels in the file plan.

Figure 6-8 shows a portion of the case study file plan that illustrates the inheritance of the disposition schedules. In one case, for insurance claims, a single disposition schedule is assigned to the Claims category and inherited at all lower levels. This particular case uses alternate retention in a single retention schedule. In the case of the Contracts category on a separate branch of the tree, one disposition schedule is assigned and inherited by the categories at the next level down. However, the Facilities category has its own disposition schedule assigned, which overrides the one from Contracts.

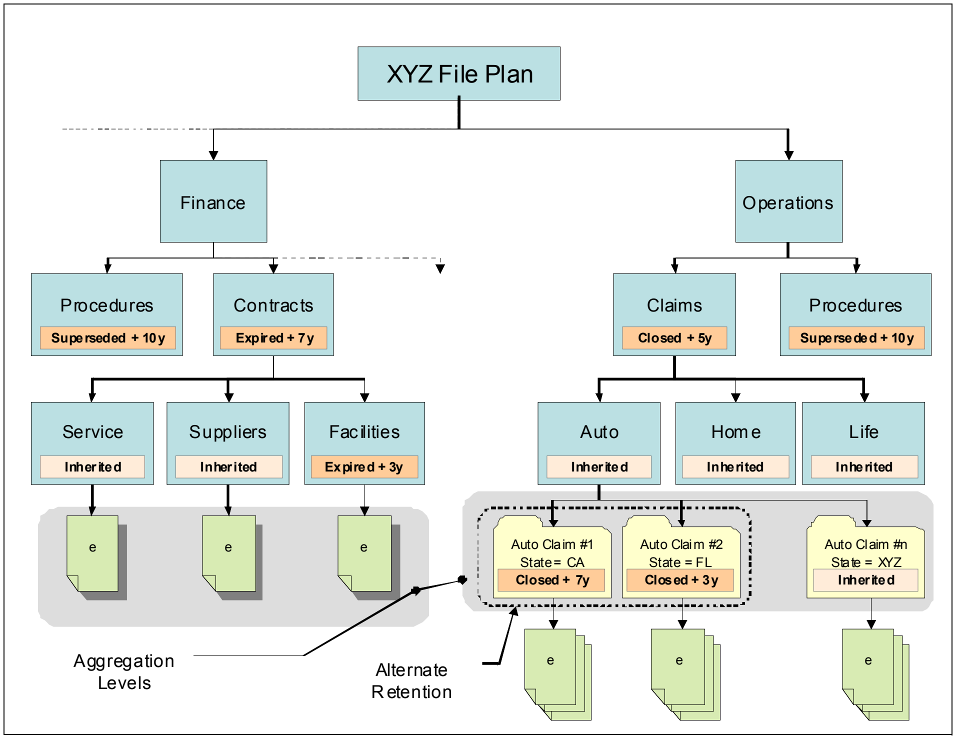


Figure 6-8 Disposition schedule inheritance and aggregation in the case study file plan

Even though a disposition schedule is inherited by all the containers in the hierarchy below where it is assigned, it only impacts those entities determined by the aggregation defined for the disposal trigger of that schedule. In other words, if a disposition schedule has a trigger with record folder aggregation, that schedule only impacts records folders; if a disposition schedule has a trigger with record aggregation, that schedule only impacts individual records. Disposition Sweep relies on the aggregation level defined by the disposal trigger to determine which entities need to be disposed.

Figure 6-8 illustrates two aggregation levels: one level for Claims and one level for Contracts. For Claims, the aggregation defined by the disposition schedule applies to records folders; while for Contracts, the aggregation defined by the disposition schedule applies to individual records. This difference impacts which entities are processed by Disposition Sweep. In the case of Claims, Disposition Sweep processes the auto claim folders, and not the individual records in each folder. Each auto claim folder is disposed of as an entity together with all the records it contains because of the aggregation specified in the disposal trigger. Contract records are disposed of individually.

The aggregation level provides flexibility in determining how you want to organize a file plan and which entities the system disposes. For example, if records are unrelated to each other aside from their common disposition schedule, it makes more sense to dispose of them independently, and it is appropriate to aggregate these records at the record level. In contrast, if records are related and must be disposed of as a group, it is more appropriate to use an aggregation level that reflects the containment object of the records. Most scenarios that involve a case folder usually call for folder-level aggregation. All of the records filed in the folder are part of that case and are retained as a unit and disposed of as a unit.

The case study is designed to illustrate the contrast between these two approaches to aggregation. The approach that you choose is determined by a variety of factors, mostly driven by your business requirements. In the case study, individual contracts are filed as individual records and therefore disposed of individually (record-level aggregation). Claims, such as auto claims, are treated as a case where the claim itself is represented as a record folder with multiple claim documents filed in each folder and the disposition schedule has folder level aggregation. Folder-level aggregation tells the system that the folder and all records in the folder are disposed of together.

These design decisions not only affect how you configure disposition schedules, but they also impact record class and record folder class design and how metadata is associated with the various classes.

Record types

In most cases, organizations design their file plan hierarchy to distinguish between various types of records and to rely on separate categories to represent the various types of records in their retention schedule. However, there are circumstances that might require records of various types to be mixed in the same container (which can be a category, record folder, or volume) while the disposition requirements are still enforced based on the type of record. In this special case, there is a feature in IBM FileNet Records Manager called *record types* that can be used to associate disposition schedules directly with individual records. Record types are designed to work with disposition schedules that have record-level aggregation. You can define the record types and associate each record type with a disposition schedule. If you want a record type to take affect, you must assign a record type to each record.

The use of record types can be combined with disposition schedule inheritance. An individual record can inherit its disposition behavior from a parent container unless it has a record type assigned, in which case the record type overrides the inherited disposition. This situation is the primary use case for record type usage: Override the default retention for individual records because of exceptional circumstances. So, record types are useful when they can be used selectively on individual records that require exceptional handling in terms of disposition.

The record types feature is an optional, advanced feature that can be useful in certain circumstances. However, in most cases, design your file plan to represent the various types of records that you plan to manage by creating categories for each type of record and locating the records or records folders in each category according to their type. This design allows you to take advantage of assigning the disposition schedule at the category level so that it can be inherited by the child entities that are to be disposed.

Best practice: Only use the *record types* feature to accommodate specific business requirements that call for mixing records with different disposition requirements in the same container or to handle exceptions on an individual basis. It is preferable to design your file plan according to your disposition requirements so records are grouped according to their disposition requirements. In addition, there is a processing overhead associated with the use of record types, because Disposition Sweep has to perform more work when calculating disposition parameters for record types. So, only use record types when the requirements demand this feature.

6.3 Disposition Sweep

Disposition Sweep is a tool that works in conjunction with the disposition schedules that you have defined for your file plan to compute essential disposition parameters and to enable disposition processing. It is a stand-alone application that is run periodically to analyze a file plan and determines which entities are ready to move through the disposition process.

6.3.1 Disposition Sweep functional overview

Disposition Sweep performs the following functions:

- Determine entities eligible for disposition
- Calculate cutoff
- Perform phase transition
- Perform phase updates
- Initiate the cutoff action
- Housekeeping (manage records)
- Process vital records
- Invoke automatic destruction

Determining entities eligible for disposition

Disposition Sweep uses the aggregation specified in the disposition schedule configuration to determine which entities are eligible for disposition processing.

Disposition Sweep performs calculations on the appropriate entities and updates the internal disposition parameters for these entities that affect their readiness for disposition processing.

Calculating cutoff

Disposition Sweep calculates the cutoff date for entities being disposed. After the condition for the cutoff trigger is satisfied, Disposition Sweep calculates the cutoff date based on either the *event date* or the *cutoff-based date* and the *offset*. For more details, refer to the "Computing cutoff" on page 158 and "Cutoff base" on page 159.

Performing phase transitions

Disposition Sweep is responsible for moving record entities from one phase to the next phase. In the case where the phase has no associated action, Disposition Sweep automatically moves the entity to the next phase. When there is an action associated with a phase, Disposition Sweep relies on the underlying workflow to move the entity to the next phase when the workflow completes.

Performing phase updates

Disposition Sweep detects if the phases of an entity's disposition schedule have been changed (for example, the retention period or action is modified) and updates the affected entities accordingly. This action also includes updates to the current phase of a disposition schedule.

Initiating the cutoff action

If the disposition schedule is configured with a cutoff action, Disposition Sweep automatically launches the associated workflow when a record entity reaches cutoff.

Housekeeping (updates)

Disposition Sweep performs a number of internal housekeeping or management chores. One task concerns resetting record entity cutoff calculations and phase transition dates when a disposition schedule has been removed from a record container. For example, if a schedule is assigned to a record category erroneously and is later removed, Disposition Sweep detects this condition and updates the affected entities accordingly.

Processing vital records

Disposition Sweep calculates the next review date for all vital records and automatically launches the associated vital records review workflow when that date has been reached or surpassed.

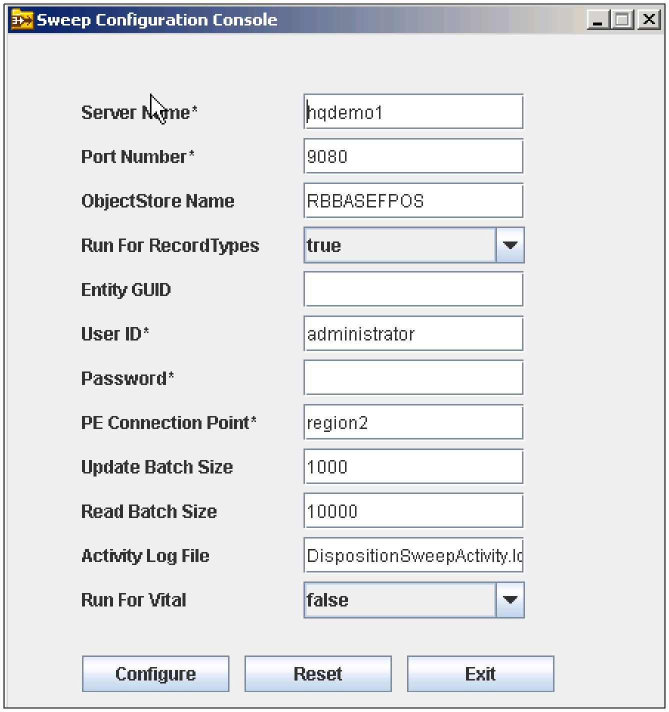
> **Note:** The processing of *vital records* is completely independent from all other Disposition Sweep functions. The calculations and actions related to vital records processing are not part of the phases of disposition.

Invoking automatic destruction

Disposition Sweep has a separate function that is used to invoke automatic destruction. By running Disposition Sweep with the appropriate command line option, Sweep will immediately begin the destruction of all entities that are ready for disposition and not on hold by using the Auto Destroy action. We discuss this topic in more detail in 6.5, “Automatic destruction (Auto Destroy feature)” on page 181.

6.3.2 Configuring Disposition Sweep

Before you run Disposition Sweep, you must configure it. Figure 6-9 shows a screen capture of the Disposition Sweep configuration console with sample parameters.



Server Name*	hqdemo1
Port Number*	9080
ObjectStore Name	RBBASEFPOS
Run For RecordTypes	true
Entity GUID	
User ID*	administrator
Password*	
PE Connection Point*	region2
Update Batch Size	1000
Read Batch Size	10000
Activity Log File	DispositionSweepActivity.lc
Run For Vital	false

Figure 6-9 Configuring Disposition Sweep

There are several parameters that must be specified. While all of these parameters are important, we describe the following parameters in more detail:

- Object Store Name: Name of File Plan Object Store (FPOS) on which you want to run Disposition Sweep. If this parameter is left blank, all object stores in the domain that are configured to be an FPOS will be processed.
- Entity GUID: Identifies a node in the file plan that limits the scope across which Disposition Sweep runs. When an entity Globally Unique Identifier (GUID) is entered, Disposition Sweep processes only that node and all children below that node. By default, this node is empty, and all entities in the FPOS are processed.
- Run for Record Types: Boolean flag that tells Disposition Sweep whether to include record types processing.
- Run for Vital: Boolean flag that tells Disposition Sweep whether to include vital records processing.

For a full description of all parameters, refer to the online help for IBM FileNet Records Manager.

The *Object Store Name* and the *Entity GUID* are useful parameters when considering deployment of Disposition Sweep (multi-instance deployments are discussed in more detail in the following section). The two parameters can be used in conjunction to limit Disposition Sweep to a particular file plan in a deployment that has multiple file plans, or they can be used to limit Disposition Sweep to a particular segment of a file plan.

Both *Run for Record Types* and *Run for Vital* options enable additional processing for Disposition Sweep and therefore add performance overhead when they are set to true. Record types and vital records are two features that are not frequently used.

> **Best practice:** For the best performance, only enable *Run for Record Types* and *Run for Vital* if you use these features. When not using these features, ensure that these parameters are set to false.

6.3.3 Deploying and running Disposition Sweep

Disposition Sweep is a stand-alone Java application that is independent of the main engines (Application Engine, Content Engine, and Process Engine) of the IBM FileNet P8 Platform. By default, Disposition Sweep is installed on Application Engine, but it does not have to be deployed on this server. Disposition Sweep is flexible in terms of how it is deployed and can either be installed on one of the main IBM FileNet P8 servers if there is sufficient capacity, or on a separate stand-alone machine. Content Engine is often a strong candidate for deploying Disposition Sweep, because having Disposition Sweep local to Content Engine can have performance benefits. However, Disposition Sweep can be installed on a completely separate machine if necessary, especially if Disposition Sweep impacts the performance of any of the underlying IBM FileNet P8 servers.

There are a number of factors that can influence how Disposition Sweep is deployed in your environment, including:

- File plan size and design
- Platform topology
- Server capacity
- Performance considerations
- Records management policies and procedures

File plan size, design, and aggregation levels influence the number of entities that Disposition Sweep needs to process. Records management policies and procedures, such as the number of file plans, how they are structured and managed, and how frequently Disposition Sweep needs to run, also strongly influence how Disposition Sweep is deployed and utilized.

It is possible to deploy multiple instances of Disposition Sweep to improve performance if necessary. You can deploy multiple instances on the same machine and or deploy instances on separate machines.

In assessing how to deploy Disposition Sweep and whether a single instance suffices or whether multiple instances are needed, consider:

- Performance: Multiple file plans or segments within the same file plan can be load-balanced across multiple servers to improve performance.
- Schedule: Multiple file plans or segments within the same file plan can have different Disposition Sweep schedules.
- Administration: Multiple file plans or segments within the same file plan can be administered by different records administrators.

Best practice: For best performance, use the Object Store Name and Entity GUID to limit Disposition Sweep to a single file plan or a portion of a file plan.

After being deployed and configured, Disposition Sweep can be run either manually from the command line or in an automated fashion using scheduling software services provided by the underlying operating system. It is typically the responsibility of the Records Administrator to schedule when and how Disposition Sweep is run in conjunction with IBM FileNet Records Manager.

Regardless of how it is run, the scheduling of Disposition Sweep is an important consideration for any enterprise. There are several important factors to consider when scheduling Disposition Sweep:

- An organization needs to determine the frequency with which to run Disposition Sweep (such as monthly, quarterly, or annually) based on the business requirements and records management policies and procedures adopted by the organization.
- On a more practical level, Disposition Sweep can have a substantial impact on system performance while running, depending on the file plan size and the number of entities that Disposition Sweep needs to process. It is therefore important to coordinate Disposition Sweep with other enterprise activities, such as system backups, so that it is run during periods where system utilization is low and impact to users is minimal.
- Disposition Sweep runs must also be coordinated with initiating disposition (which we discuss in the next section).

Disposition Sweep logs all of its activity, including any errors that it detects during its processing, to a specified activity log file. This information can be reviewed by the Records Administrator to determine how Disposition Sweep is performing and can be used to fine-tune the configuration for optimal performance.

6.3.4 Performance considerations

The file plan design, file plan size, and how you choose to aggregate for purposes of disposition have a direct impact on the performance of Disposition Sweep. Aggregation is an extremely influential factor in determining how many entities Disposition Sweep needs to process any time that it is run.

In general, the more often that you can aggregate at the container level (either record folder or volume), the fewer items Disposition Sweep has to process. There are use cases and scenarios where it is impractical to aggregate at the container level. In these cases, be aware that aggregating at the record level requires that you develop the appropriate strategies for deploying Disposition Sweep to handle the anticipated load.

There are also general performance tuning practices to follow for IBM FileNet Records Manager. For example, by creating the appropriate indexes for system and custom properties in the database, you can greatly improve Disposition Sweep performance. The specific properties that require indexing can depend upon the metadata that you choose for your disposal trigger criteria. Refer to the *IBM FileNet P8 Performance Tuning Guide*, GC31-5483, for more details.

Best practice: Work with your Database Administrator (DBA) in the early stages of implementation to properly tune the database for IBM FileNet Records Manager. Become familiar with various performance tuning criteria that are affected by your file plan design and your specific business requirements.

6.4 Initiating and completing disposition actions

Initiating disposition is the primary mechanism used to launch the workflows that implement the disposition actions. The workflows define the steps required to complete the disposition actions.

We summarize the events leading up to initiating disposition:

- A trigger condition has been satisfied (for example, an insurance claim has been closed or a contract expiration date has been set).
- Disposition Sweep has run, and cutoff is calculated.
- Cutoff is achieved (with or without the optional cutoff action), and the entity is transitioned to the first phase of disposition.
- The retention period for the disposition phase has elapsed.

After the retention period has elapsed for an entity, it is ready for disposition, meaning that the action associated with the disposition phase is ready to be initiated.

6.4.1 Initiating disposition

Initiating disposition can be invoked manually from the IBM FileNet Records Manager Web application. In most cases, disposition is initiated at the higher levels of the file plan by invoking the Initiate Disposition command on the desired sections of the file plan. The command cascades down the file plan hierarchy and aggregates entities in batches for disposition processing. It is typically the role of the Records Manager to initiate disposition as appropriate based on business requirements and records management policies and procedures for the organization.

At any one time, there can be many thousands of entities ready for disposition scattered all over the file plan. It is a rather daunting task to search for individual entities and selectively initiate their disposition. Instead, the Records Manager must rely on the design of the file plan hierarchy to initiate disposition over entire categories, allowing the Initiate Disposition command to automatically aggregate entities in batches.

The Initiate Disposition command ignores any entity that is not ready for disposition, which includes any entity that is not configured for disposition, any entity for which the retention period has not yet ended, or any entity placed on hold.

Strategies for initiating disposition

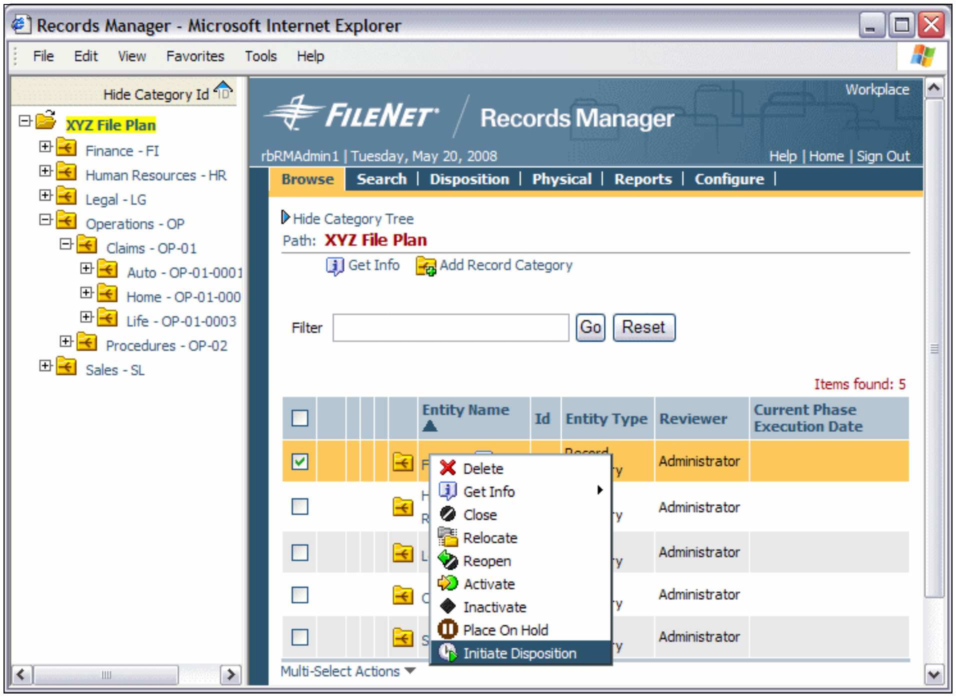
A Records Manager can either browse or search for categories in the file plan upon which disposition must be initiated. In a well-designed file plan, a typical approach is to browse to the first level of the file plan and select the desired categories representing the various business functions at level one. Whether to initiate across the entire file plan at once or to be selective in initiating a single category at a time is dependent upon the size of the file plan, the number of entities expected to be ready for disposition, the performance capacity of the system, and on the specific business requirements at any given time.

In our case study file plan, for example, you might choose to initiate disposition on the top-level business function categories, such as Finance, on a quarterly or annual basis in order to dispose of all financial records for that time period that are ready for disposition. In contrast, you might choose to browse or search for specific categories at a lower level in the file plan, such as Service, Suppliers, or Facilities categories under Contracts, and initiate disposition on each of these categories separately. In a production file plan, this approach can be too burdensome because of the sheer number of categories at this level. The practical approach taken depends on the file plan design and your own business practices and records management policies and procedures.

Figure 6-10 shows the Initiate Disposition command being selected from a context menu in the IBM FileNet Records Manager Web application browse interface. In this example, the command is being applied to a single entity, but the command is also available as a multi-select action where it can be applied to multiple entities at one time.

Best practice: Records Managers must develop a strategy and plan for manually initiating disposition based on the business requirements and records management policies and procedures of the organization. Initiating disposition must be coordinated with other activities, such as running Disposition Sweep, applying record holds, and updating the disposition schedules to reflect the latest laws, regulations, and policies that determine your retention rules.

In rare cases, it might be necessary to search for and initiate disposition on individual entities that are ready. However, only use this approach in special circumstances. It is typically a records management best practice to routinely initiate disposition across an entire category, allowing the system to aggregate entities into batches automatically.



□	Entity Name	Id	Entity Type	Reviewer	Current Phase Execution Date
☑	F		Record ... y	Administrator	
□	H R		y	Administrator	
□	L		y	Administrator	
□	O		y	Administrator	
□	S		y	Administrator	

Figure 6-10 Initiating disposition from the browse page on a single category

Automating the initiate disposition function

The IBM FileNet Records Manager API exposes the appropriate methods for initiating disposition from a stand-alone utility. If your business requirements and file plan design dictate a more automated approach, you can develop your own utility to select entities for disposition and initiate disposition as needed.

6.4.2 Disposition processing in batches

Initiating disposition in a production environment typically affects large numbers of record entities at any one time. You typically do not want to launch individual disposition workflows for every entity and process each entity individually. It is more efficient to let the system create batches of related entities to be processed as a group.

Batch size

IBM FileNet Records Manager allows you to configure the batch size for disposition workflows. This parameter is a system-wide configuration parameter that applies to all disposition workflows.

Batch size ranges from 1 to 500 entities per batch. The default batch size is 10. The batch size that you select is primarily determined by your business requirements.

Criteria for assembling batches

When initiating disposition, the system assembles batches of entities that are ready for disposition based on the batch size. The system also uses the following two criteria to group entities into separate batches:

- Reviewer: The reviewer property value assigned to the entity to be disposed
- Action: The specific action assigned to the disposition phase being initiated

The reviewer property is a required property for all record entities. The reviewer property is typically set to the user who declares the record or who creates a container. However, you can set this property as part of your record declaration process to control how records are batched for disposition, or you can update the property before you initiate disposition in order to control how entities are batched. A single batch only includes entities that are assigned the same reviewer.

Best practice: Assign specific values to the reviewer property during the record declaration process to control how records are grouped into batches for disposition processing. The person who declares a record might not necessarily be the most appropriate person to review the record during disposition.

The action that you assign to a disposition phase is also used to control the batch groupings. A single batch only includes entities that are assigned the same action. You can use this feature to your advantage to control batching to a certain degree by either configuring separate actions for different disposition schedules or by reusing the same action across multiple disposition schedules.

6.4.3 Completing the disposition process

A disposition phase is completed when the workflow associated with the action for that phase has completed. The process of completing the steps of the workflow is entirely controlled by the workflow itself. Each disposition workflow has specific steps that are completed according to the specified process and the action to be performed. These actions are described in more detail in 6.2.4, “Disposition phases and actions” on page 160. You can customize the predefined workflows that are provided to meet specific business requirements.

6.5 Automatic destruction (Auto Destroy feature)

A new feature introduced with the IBM FileNet Records Manager 4.5 release is the ability to configure disposition for automatic record destruction. In previous releases, record destruction is performed by a managed and configurable process that relies on workflows to complete the destruction. While most disposition use cases still call for these workflow-based and configurable destruction processes that typically require user approval and review, there are specific use cases that are better suited to automatic destruction.

6.5.1 When to use Auto Destroy

The Auto Destroy feature is typically implemented for use cases where no approval is required prior to record destruction and where there is a large volume of records being managed and destroyed by the system. A simple e-mail archiving solution might be this type of a use case, where hundreds of thousands of e-mails are ingested every day and e-mails are only retained for a short period of time, for example, 90 days. Unless the e-mail records have been placed on hold during the retention period, you might want to immediately destroy the e-mails when the retention period has elapsed. This type of a use case typically does not require any review or an elaborate process during disposition.

6.5.2 Advantages of Auto Destroy

Auto Destroy provides the following advantages:

- Avoid the overhead of launching and completing a workflow.
- Avoid the need to customize the destroy workflow.
- Achieve faster processing by directly destroying entities instead of relying on workflow and component integration to complete the record destruction.

6.5.3 Configuring a disposition schedule for automatic destruction

In order to use Auto Destroy, you must configure an Auto Destroy action, which is described in “Auto Destroy” on page 164. The Auto Destroy action is not linked to any workflow, but it can be used like any other disposition action as a phase in a disposition schedule. Remember that any destroy phase is the final phase in a disposition schedule. After an Auto Destroy action is configured, it can be used in a disposition schedule.

6.5.4 Executing automatic destruction

After you configure a disposition schedule to use an Auto Destroy action, the automatic destruction is executed using Disposition Sweep. First, any entities that are controlled by a disposition schedule with an auto destroy phase must be processed by Disposition Sweep in the usual fashion to compute the disposition date values, allowing the entities to be ready for disposition at the appropriate time. An additional step is then required to execute the automatic destruction. This step is performed as a separate command line option for the Disposition Sweep tool. When this command is invoked, the system immediately and automatically destroys all entities having the Auto Destroy action that are ready for disposition and that are not on hold. And just as with the regular Destroy action, the Auto Destroy action treats entities that are filed in multiple containers the same way; namely if an entity is filed in multiple containers, it will only be unfiled and is not destroyed until it is removed from the last parent container.

7

Records hold

In the event of litigation, investigations, or auditing actions, organizations must be able to prevent the destruction of pertinent records. Destruction can happen during records disposition, or it can be manually initiated by a Records Manager. By placing records on hold, the Records Administrator can insure that they are not destroyed. When a record is placed on hold, it cannot be destroyed either through the normal disposition process or even manually until the hold is removed. In this chapter, we discuss hold processing in IBM FileNet Records Manager.

We describe the following topics in this chapter:

- Definition of hold
- Hold processing in IBM FileNet Records Manager
- Performance considerations

7.1 Definition of hold

A *hold* is an action taken on records or containers to guarantee that they are not disposed of in accordance with their defined retention rules. Placing a record or container on hold stops the disposition process for that record to insure that it is available for pending actions. When the pending action is resolved, the hold can be removed from the record or container, which restarts the disposition process.

In order to prevent the accidental deletion of the record, a hold also prevents the record from being manually deleted, which gives the Records Administrator confidence that records that are placed on hold will be available until the pending action is resolved and the hold is removed.

Many laws and regulations govern the retention of records across a wide range of industries and geographies. Primary regulators include the Securities and Exchange Commission (SEC), the Food and Drug Administration (FDA), the Department of Labor, and the Environmental Protection Agency (EPA). The retention regulations advocated by these agencies must be taken into consideration when creating records retention schedules and policies, because these organizations mandate that records are preserved until a specified period of time has passed or an event occurs. After this time has passed or event happens, the records are eligible for destruction, *except when the records are the subject of potential or pending litigation or investigation*.

Note: *Freeze* is an alternative term for hold. The Department of Defense (DoD) specification uses the term *freeze*.

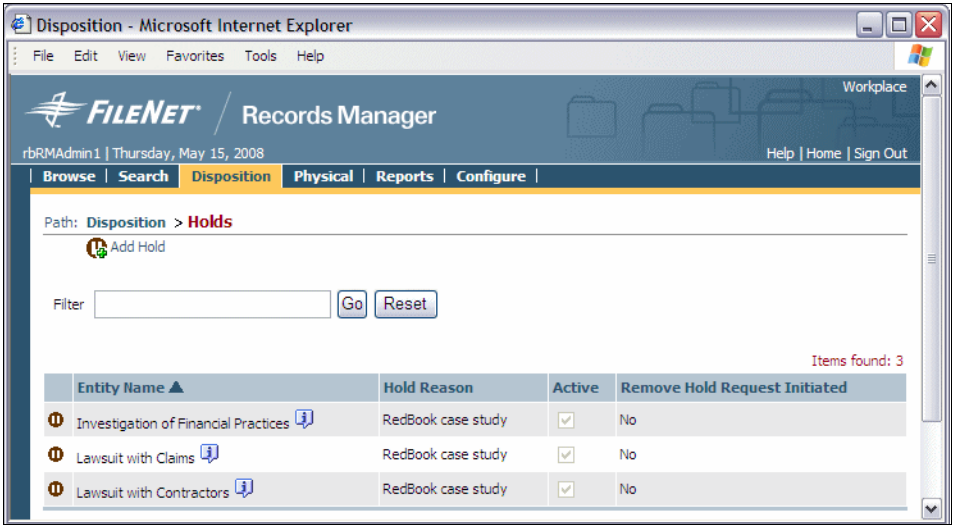
7.2 Hold processing in IBM FileNet Records Manager

IBM FileNet Records Manager allows Records Administrators or Records Managers to identify relevant entities, such as categories, folders, volumes, or records, and manually place holds on them. Alternatively, the placing of holds on entities can be automated through the use of conditional (dynamic) holds. By default, only users and groups assigned the Records Administrator or Records Manager role can create and apply holds.

Holds are automatically inherited by lower levels of the file plan. Using our case study as an example, if a hold is placed on the record category Operations, all categories, folders, volumes, and records under Operations are automatically on hold. IBM FileNet Records Manager prevents anyone from overriding this hold until the hold is removed.

When a hold is placed on an entity, the hold overrides normal disposition processing. The hold prevents an authorized user from initiating disposition on the entity until the hold is removed. In addition, all disposition actions are suspended until the hold is removed. Putting a hold on an entity does not prevent Disposition Sweep from calculating the *Cut Off Date* or the *Current Phase Execution Date* for that entity. After the hold is removed, the entity is again eligible for disposition if the *Current Phase Execution Date* has been surpassed.

When a hold is created, you have to specify whether it is active or inactive. Refer to Figure 7-1. Only active holds can be placed on records. A Records Administrator can define a hold at any time. However, if the hold is not used immediately but at a later point, or a Records Administrator determines that a hold is no longer to be used, the hold is set to inactive.



Path: Disposition > Holds

Add Hold

Filter Go Reset

Items found: 3

	Entity Name ▲	Hold Reason	Active	Remove Hold Request Initiated
	Investigation of Financial Practices	RedBook case study	☑	No
	Lawsuit with Claims	RedBook case study	☑	No
	Lawsuit with Contractors	RedBook case study	☑	No

Figure 7-1 Active and non-active holds

Audit and legal holds

When creating a hold, you can define whether the hold is for audit or legal purposes.

A *legal hold* is used when a company must take measures to make sure that records, which are the subject of pending or potential *litigation or investigation,* are not destroyed until the litigation or investigation is over, that is, until after the legal hold is lifted.

An *audit hold* is used when a company must prevent the deletion of records that are the subject of internal or external auditing, attestation, or quality control processes.

Functionally, these holds are exactly the same; IBM FileNet Records Manager does not process them differently. The reason for specifying the hold type as Audit or Legal is informational only.

If required, you can alter or add to these values. The values are defined in a Choice List, **HoldTypeList**, in the File Plan Object Store (FPOS). To update these values, edit the choice list using Enterprise Manager.

7.2.1 Manual holds

A hold created with no conditions is a *manual hold*. A manual hold is designed for dynamic use where the Records Administrator or Records Manager will search or browse for records to include in the hold. They then manually place these records on hold. Typically, manual holds are only used for placing a small number of specific entities on hold. For instance, if you have a folder associated with a specific claim, you can manually place the folder on hold if there is an open action about that claim.

7.2.2 Dynamic holds

Dynamic holds, which are also known as *conditional holds,* are used to address the situation where a Records Manager needs to put a large number of records on hold, and these records are potentially distributed throughout the file plan. In addition, a dynamic hold can be run on a regular basis to identify new documents that satisfy the hold conditions. With dynamic holds, you create search conditions to indicate which entities must be placed on hold. If there are no search conditions for a hold, that hold can only be added to an entity manually.

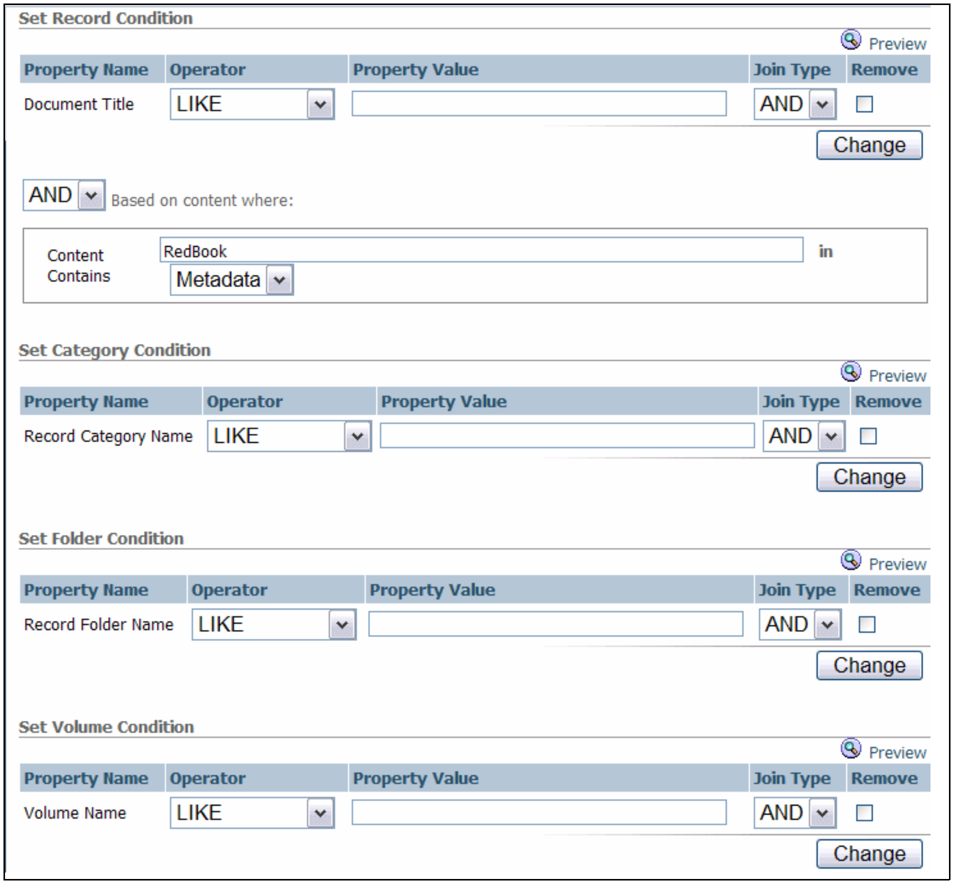
After the dynamic hold is created, it is applied by running the Hold Sweep application. This application searches the records repository and identifies each entity that needs to be placed on hold. It then applies the hold to each of those entities. Over time, new records can be declared that meet the hold criteria. In this case, Hold Sweep can be run on a regular basis, and it automatically places holds on the new records that meet the hold conditions.

The search conditions are created using the metadata on the entity. Refer to Figure 7-2 on page 187. The search conditions can differ for each type of entity (category, folder, volume, or record) and can use up to five pieces of metadata per entity, which are joined through the logical operators AND and OR. Additionally, content-based retrieval of the content associated with a record can be leveraged as part of the search condition.

If search conditions are specified for multiple entity types, they are treated separately. For example, if the conditional hold is created with the following search conditions:

- For records with `XYZClaimType = Auto`
- For record categories with `Date Created <= 01/01/2006`

The search does not search for records with XYZClaimType = Auto that exist in a record category that is created after 01/01/2006. It searches for all records with XYZClaimType = Auto and for all record categories that are created on or before 01/01/2006.


Set Record Condition

Property Name	Operator	Property Value	Join Type	Remove
Document Title	LIKE		AND	□

AND Based on content where:

Content Contains: RedBook in Metadata

Set Category Condition

Property Name	Operator	Property Value	Join Type	Remove
Record Category Name	LIKE		AND	□

Set Folder Condition

Property Name	Operator	Property Value	Join Type	Remove
Record Folder Name	LIKE		AND	□

Set Volume Condition

Property Name	Operator	Property Value	Join Type	Remove
Volume Name	LIKE		AND	□

Figure 7-2 Building search conditions for dynamic hold

7.2.3 Multiple holds

Typically, an organization has multiple litigation actions or audits occurring at the same time, with the potential that an entity might be discovered and placed on hold for each action or audit, resulting in an entity having multiple holds on it at the same time.

For instance, suppose a company receives a letter from an opposing counsel or investigating agency to obtain discoverable records. Refer to Figure 7-3. This action automatically triggers hold 1 on a record.

In this case, we guarantee that records will not be destroyed until the end of trial process 1.

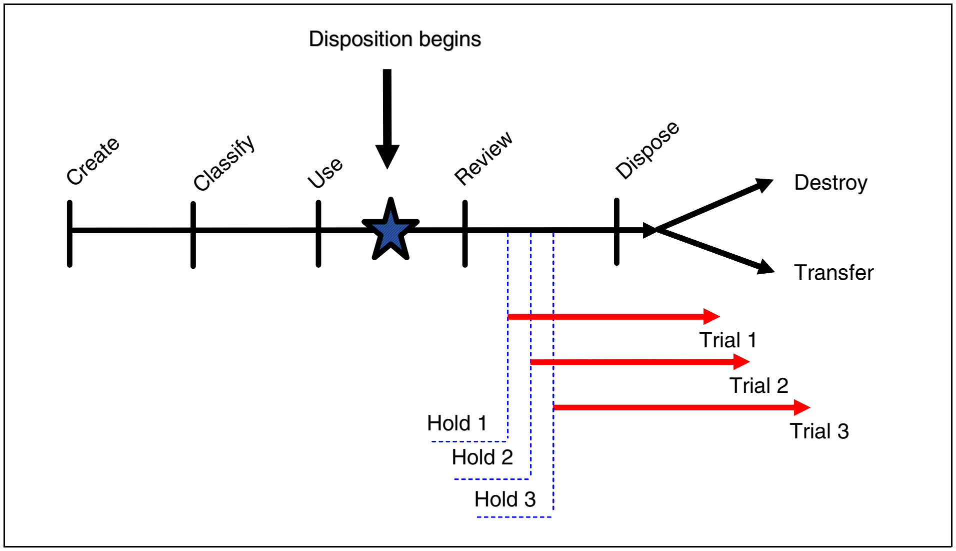


Figure 7-3 Multiple holds needed scenario

What happens if another trial begins against the same record? Another hold is added to the same entity. In our example, three holds are added to the same record for three separate legal matters.

Each of these holds is removed at a different time depending on when the litigation is over. Only when the final hold is removed will the entity be eligible for disposition.

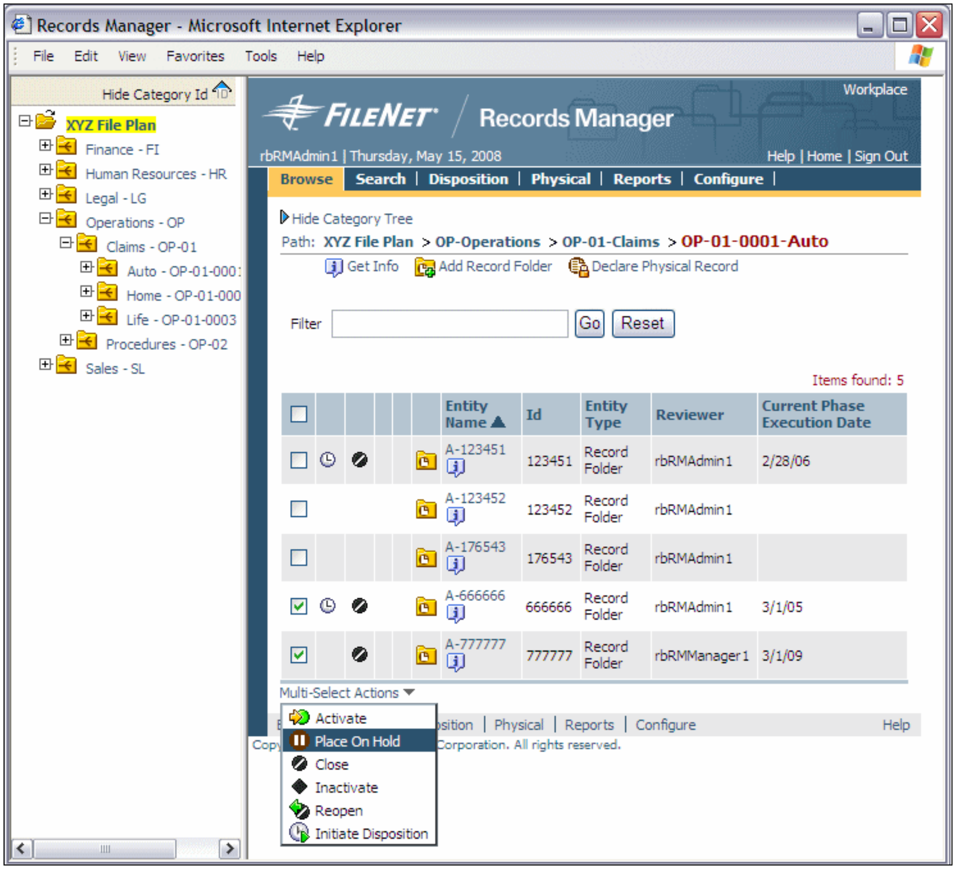
7.2.4 Applying holds

You can apply holds manually or automatically.

Placing a hold manually

Using the Browse page (Figure 7-4), a Records Manager can navigate the file plan to identify entities to be placed on hold.

To place an entity on hold manually, right-click the **Get Info** icon next to the appropriate entity, which opens a menu, and select **Place On Hold**.



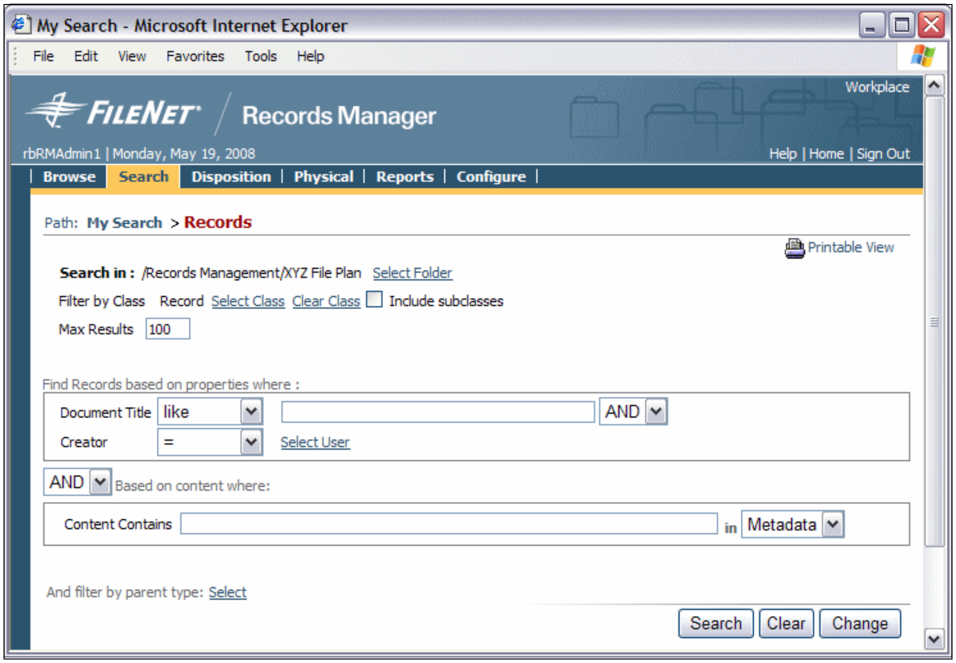
Path: XYZ File Plan > OP-Operations > OP-01-Claims > OP-01-0001-Auto

Items found: 5

☐				Entity Name ▲	Id	Entity Type	Reviewer	Current Phase Execution Date
☐	🕓	⊘		A-123451	123451	Record Folder	rbRMAdmin1	2/28/06
☐				A-123452	123452	Record Folder	rbRMAdmin1	
☐				A-176543	176543	Record Folder	rbRMAdmin1	
☑	🕓	⊘		A-666666	666666	Record Folder	rbRMAdmin1	3/1/05
☑		⊘		A-777777	777777	Record Folder	rbRMManager1	3/1/09

Figure 7-4 Browse page: Manually put records on hold

The Search page (Figure 7-5 on page 190) enables you to search for specific entities that need to be placed on hold. From the returned result set, you can perform individual or bulk operations on the entities. For instance, you can search for all records from a particular customer and place a hold (or holds) on all these records in one operation using the **Place On Hold** function from within the **Multi-Select Actions** menu.


My Search - Microsoft Internet Explorer
File Edit View Favorites Tools Help
FileNet Records Manager
Workplace
Help | Home | Sign Out
Browse | Search | Disposition | Physical | Reports | Configure
Printable View
Select Folder
Select Class
Clear Class
Select User
Search
Clear
Change

rbRMAdmin1 | Monday, May 19, 2008

Path: My Search > Records

Search in : /Records Management/XYZ File Plan

Filter by Class Record ☐ Include subclasses

Max Results 100

Find Records based on properties where :

Document Title like AND

Creator =

AND Based on content where:

Content Contains in Metadata

And filter by parent type: Select

Figure 7-5 Search page: Search through file plan for records to be put on hold

Placing a hold automatically

Holds can be placed on records automatically by using the defined search conditions in conjunction with the Hold Sweep program. Refer to 7.2.6, “Hold Sweep” on page 193 for full details.

7.2.5 Removing holds

Similar to placing holds on records, you can remove holds on records manually or automatically.

Removing holds manually

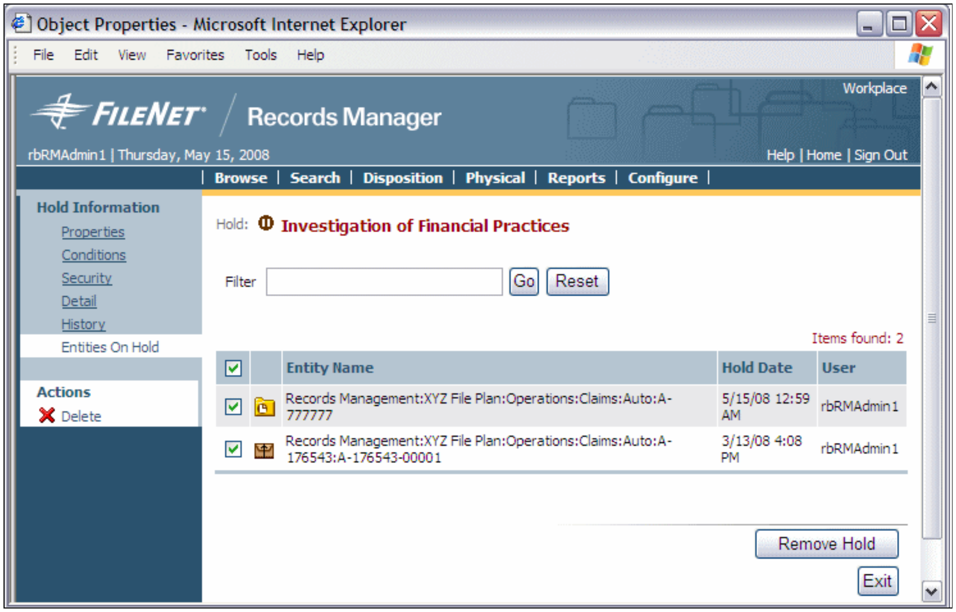
To manually remove holds, IBM FileNet Records Manager offers three mechanisms:

- Browsing the file plan using the Browse page (refer to Figure 7-4 on page 189)
- Searching directly the held records using the Search page (refer to Figure 7-5)
- Using the Disposition page feature (refer to Figure 7-6 on page 191)

To search for the held records directly, in the second method, you can search for entities whose On Hold property value is equal to True.

For the first two options, after the appropriate records have been identified, go to the Information Page for each entity, select **Holds** from the menu at left side of the window, and one or more holds will appear. Select them and click **Remove Hold**.

Using these options, you are not allowed to remove a hold from multiple records at the same time.


Object Properties - Microsoft Internet Explorer
File Edit View Favorites Tools Help
FileNet Records Manager
Workplace
rbRMAdmin1 | Thursday, May 15, 2008
Help | Home | Sign Out
Browse | Search | Disposition | Physical | Reports | Configure
Hold Information
Properties
Conditions
Security
Detail
History
Entities On Hold
Actions
Delete
Go
Reset
Remove Hold
Exit

Hold: Investigation of Financial Practices

Filter

Items found: 2

☑		Entity Name	Hold Date	User
☑		Records Management:XYZ File Plan:Operations:Claims:Auto:A-777777	5/15/08 12:59 AM	rbRMAdmin1
☑		Records Management:XYZ File Plan:Operations:Claims:Auto:A-176543:A-176543-00001	3/13/08 4:08 PM	rbRMAdmin1

Figure 7-6 Entities on hold through Disposition → Hold → Entities On Hold

The most likely scenario is that the litigation or audit is complete and the associated hold needs to be removed from all records that are related to the specific litigation or audit. To accomplish this hold removal, use the third option, through the Disposition page:

1. Select the **Disposition** tab.
2. Select **Holds** from the left menu.
3. From the Hold window, select the **Get Info** icon next to the hold that needs to be removed.
4. Select **Entities on Hold** from the left menu. Refer to Figure 7-6.
5. Select individual entities or all entities (subject to the maximum number per page imposed by Workplace), and click **Remove Hold**.

Note: Manual holds, that is, holds without conditions, can only be removed manually.

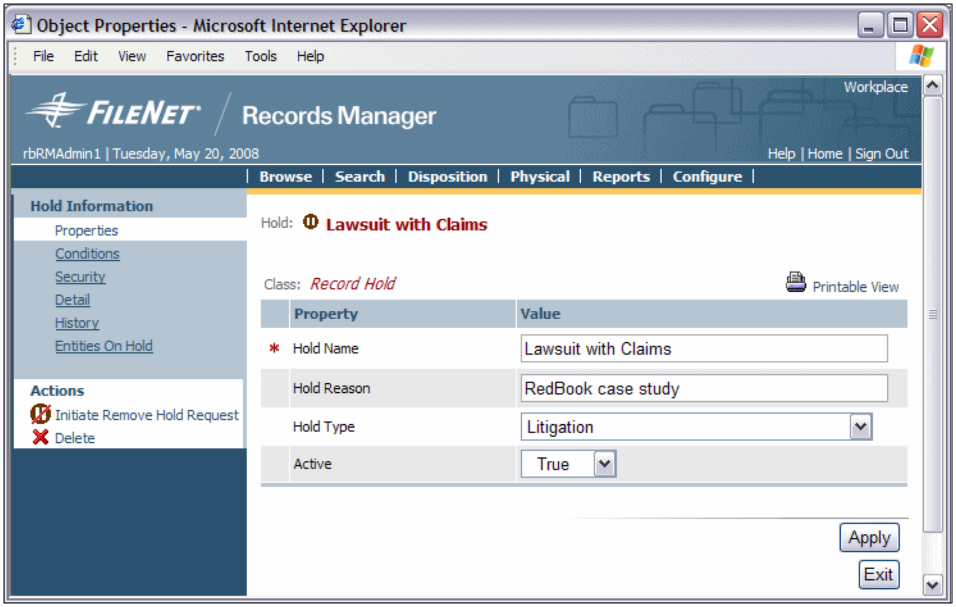
Removing holds automatically

Holds that are placed on entities by running Hold Sweep can only be removed by running Hold Sweep again after initiating a remove hold request (refer to Figure 7-7).

To initiate removing a hold request:

1. Select the **Disposition** tab.
2. Click **Holds** from the left menu.
3. Right-click the hold, and select **Initiate Remove Hold Request**. The Remove Hold Request page asks if you want to remove this hold from associated entities in the next Hold Sweep run.
4. Click **Accept** to remove the hold, or click **Exit** to close without initiating the request.

The next time that Hold Sweep runs, all entities that have been associated with this hold will have this hold removed. This action includes entities that have this hold placed on them manually.



Hold: Lawsuit with Claims

Class: *Record Hold*

	Property	Value
*	Hold Name	Lawsuit with Claims
	Hold Reason	RedBook case study
	Hold Type	Litigation
	Active	True

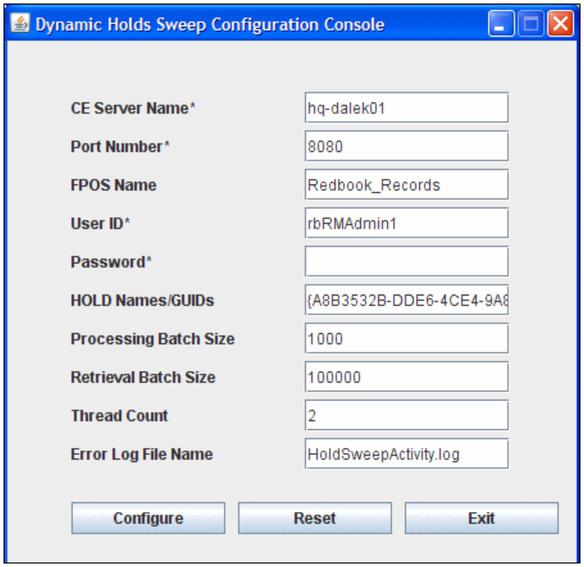
Figure 7-7 Initiate remove hold request

7.2.6 Hold Sweep

Hold Sweep is an application that is responsible for finding records that meet the conditions specified in dynamic holds and for placing holds on these records. Hold Sweep is also responsible for removing the holds when the litigation action or the audit is complete.

Configure Hold Sweep

Before you can run Hold Sweep, you must configure it to specify which hold in which FPOS it is to run. Refer to Figure 7-8.



Dynamic Holds Sweep Configuration Console

CE Server Name*	hq-dalek01
Port Number*	8080
FPOS Name	Redbook_Records
User ID*	rbRMAdmin1
Password*	
HOLD Names/GUIDs	{A8B3532B-DDE6-4CE4-9A8
Processing Batch Size	1000
Retrieval Batch Size	100000
Thread Count	2
Error Log File Name	HoldSweepActivity.log

Figure 7-8 Dynamic Holds Sweep Configuration Console

To configure Hold Sweep:

1. From a command prompt on the machine where you installed Hold Sweep, navigate to the RecordsManagerSweep folder. Enter one of the following commands:
   - For Windows®: **`RecordsManagerSweep.bat -HoldSweep -configure`**
   - For Unix: **`./RecordsManagerSweep.sh -HoldSweep -configure`**

   The Dynamic Hold Sweep Configuration Console dialog box appears, which is shown in Figure 7-8.

2. Specify the appropriate values for the following fields. The fields with an asterisk (*) are required (clear existing values by clicking Reset):

   - CE Server Name: This field is the name or IP address of the Content Engine server.
   - Port Number: This field provides the Web Services Interoperability (WSI) port number used by your Content Engine server. For example, the default port number for Content Engine running under: WebSphere® is 9080, WebLogic is 7001, and JBoss® is 8080.
   - FPOS Name: This field is the Globally Unique Identifier (GUID) or the name of the File Plan Object Store in which you want to run Hold Sweep. If you do not provide a value, the Hold Sweep process runs on all of the File Plan Object Stores associated with the specified Content Engine server. If the name of the object store contains extended characters, use the GUID instead of the name.

     GUID is the Globally Unique Identifier (GUID) of the IBM FileNet P8 domain. Every Content Engine object has a GUID, and it cannot be changed.
   - User ID: This field is the user name that Hold Sweep uses to log on to Content Engine to perform calculations. The user must have object store administrative rights in the FPOS and possess Records Administrator privileges.
   - Password: This field is the password for the user ID. The password must be specified each time that a change is made to the configuration.
   - Hold Names/GUIDs: Name or GUID of up to five holds, separated by the (|) character. The Hold Sweep process uses only the specified holds. If no hold is provided, Hold Sweep processes all the active holds.
   - Processing Batch Size: This field is the number of entities to be processed as a batch using the Hold Sweep process. By default, this value has been set to 1000. For example, if this value is 1000 and there are 20,000 entities to be processed, Hold Sweep processes all entities in 20 batches, with 1000 entities in each batch.
   - Retrieval Batch Size: This field is the number of entities to be retrieved per batch using the Hold Sweep process. By default, this value has been set to 100,000. For example, if this value is 100,000 and there are 1,000,000 entities to be processed, all the entities are retrieved in 10 batches, with 100,000 entities in each batch.
   - Thread Count: This field is the number of threads to be used for hold processing. Typically, this value must match the number of processors on the server where Hold Sweep is running, but the value needs to be adjusted based on tuning the system.

- Error Log File Name: This field is the name and path of the error file to be created by the Hold Sweep process, or you can accept the default. By default, a file called HoldSweepActivity.log is created in the ../FileNet/RecordsManagerSweep folder.

3. Click **Configure**. You will see a message indicating the successful configuration of Hold Sweep.

Running Hold Sweep

The execution of Hold Sweep depends on customer requirements. You run it when you need to automatically add or remove holds. In order to avoid an impact on system performance, always run Hold Sweep when your *system usage is low*.

To run Hold Sweep, from a command prompt on the machine where you installed Hold Sweep, navigate to the RecordsManagerSweep folder. Enter one of the following commands:

- For Windows: **`RecordsManagerSweep.bat -HoldSweep`**
- For Unix: **`./RecordsManagerSweep.sh -HoldSweep`**

Verify whether Hold Sweep ran successfully by viewing the error log file created in the RecordsManagerSweep folder. If the error file is empty, the Hold Sweep process ran successfully. Otherwise, the file contains errors that you can use to troubleshoot the problem.

Depending on the number of entities that need to be processed, Hold Sweep can take a considerable amount of time to run, and it might impact normal business operations. When you need to stop the Hold Sweep processing, add the **`-stop`** parameter in the command prompt:

- For Windows: **`RecordsManagerSweep.bat -HoldSweep -stop`**
- For Unix: **`./RecordsManagerSweep.sh -HoldSweep -stop`**

When you are ready to run Hold Sweep again, you can execute the normal **`HoldSweep`** commands without the **`-stop`** parameter.

7.2.7 Inheritance of holds

A hold can be placed directly on a record, or it can be set at a point in the file plan. If it is set on an individual record, only the record on which the hold is placed is on hold. If it is set at a point in the file plan (such as a category or a folder), all records filed below that point in the file plan hierarchy inherit the hold.

7.2.8 Impact on holds of the disposal trigger aggregation level

A factor that influences hold functionality is the aggregation level of the disposition schedule's internal trigger that applies to the entity. If a record is put on hold, but the disposition of the record is aggregated at a higher level (folder or category), the hold on this record prevents the entire folder or category from being disposed. The implication is that an organization might in fact have a greater number of records on hold than they are really aware of, resulting in records that they thought were eligible for disposition but not being disposed. If you want to avoid this behavior, evaluate the impact of aggregation at the record level.

7.3 Performance considerations

Hold Sweep is a resource-intensive operation from a database perspective. For this reason, it is best to schedule Hold Sweep to run when system usage is low. To set the correct expectation, if you have a conditional hold that might put on hold approximately a million records, Hold Sweep can take several hours to complete.

The Hold Sweep configuration has a processing batch size (set to 1,000) and a retrieval batch size (set to 100,000). These values have been sized for production use. Only change them after you thoroughly examine and understand the implications of changing them.

For instance, the 100,000 batch size for retrievals means that the Java virtual machine (JVM™) that is running Hold Sweep needs at least 1 GB of heap memory to be able to process the result set, and the *QueryPageMaxSize* setting in the Server Cache Configuration of IBM FileNet Enterprise manager is set at a value greater than this batch size.

Increasing the thread can increase the throughput of the hold process at the expense of an increased processor load. Perform tuning to determine the optimum thread count for your environment; however, never set the thread count higher than the number of processors on the server where the hold is running.

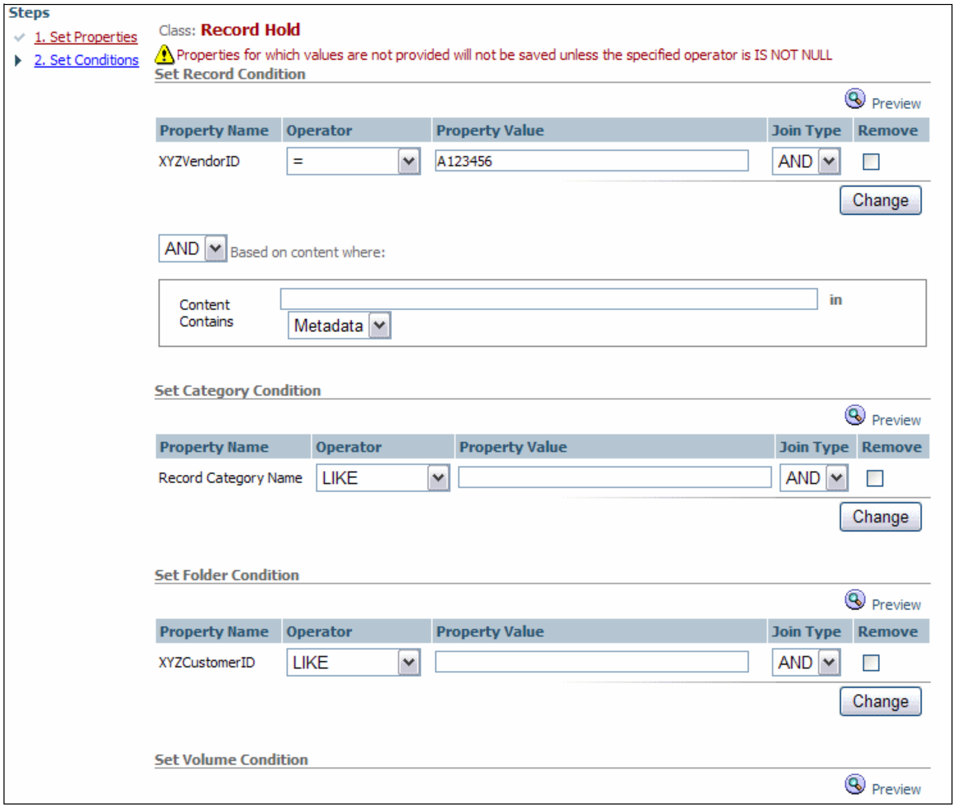
Unlike Disposition Sweep, the size and depth of a file plan have no affect on the performance of Hold Sweep. Hold Sweep uses the conditions defined in the hold definition to directly search for entities no matter where they reside in the file plan. It is therefore important to ensure that these searches run optimally, which means that the best practice is to routinely tune the underlying database.

A best practice is to create database indexes for the metadata elements that are expected to be searched frequently. In addition, create the database indexes on two system properties that are defined on entities: *On Hold* and *Prevent RM Entity Deletion*.

The searches are also ordered by GUIDs. It is therefore critical for large systems to build a cluster index on GUID to improve performance.

It is worth pointing out here that the more complex the condition, for example, with multiple attributes or content-based retrieval, the greater the impact on hold performance and the longer that it takes to run the hold.

If a hold contains conditions for separate, such as records, folders, and categories, there are three independent searches, which Hold Sweep performs sequentially. For example in the Figure 7-9, a hold is being created for a litigation action that is underway against a contractor to Fictional Insurance Company X. This contractor has both contracts and claim documents that need to be put on hold, so conditions have been created to search for both records and folders that have the contractor's ID in the respective VendorID or CustomerID metadata fields.



Steps

1. Set Properties

2. Set Conditions

Class: Record Hold

⚠ Properties for which values are not provided will not be saved unless the specified operator is IS NOT NULL

Set Record Condition

Preview

Property Name	Operator	Property Value	Join Type	Remove
XYZVendorID	=	A123456	AND	□

Change

AND Based on content where:

Content Contains ______ in Metadata

Set Category Condition

Preview

Property Name	Operator	Property Value	Join Type	Remove
Record Category Name	LIKE		AND	□

Change

Set Folder Condition

Preview

Property Name	Operator	Property Value	Join Type	Remove
XYZCustomerID	LIKE		AND	□

Change

Set Volume Condition

Preview

Figure 7-9 Multiple hold conditions

When Hold Sweep is run, it treats the two search conditions as independent searches and runs and acts on them sequentially. In this example, it first searches for and place on hold all records where XYZVendorID=A123456, and it then searches for and places on hold all folders where XYZCustomerID=A123456.

Finally, as a general rule, it takes Hold Sweep the same amount of time to remove holds from entities as it takes to put the entities on hold initially.

8

Auditing and reporting

In this chapter, we introduce the requirements for auditing and reporting within a records management system. We also describe the extensions to the built-in auditing framework provided by IBM FileNet Records Manager and how this audit information can be accessed and reported.

We discuss the following topics in the chapter:

- Introduction to auditing
- Auditing an IBM FileNet Records Manager system
- Reporting
- Performance implications

8.1 Introduction to auditing

The International Standardization Organization (ISO) 15489 standard for records management mandates that the ability to audit a records management system is a fundamental requirement. There are two major reasons why auditing is required:

- To ensure compliance with the organization's standards
- To ensure that records will be accepted as evidence in a court of law

Compliance auditing

A requirement of a record management system is to provide evidence of an organization's:

- Understanding of the nature of its records
- Care and security arrangements for the records
- Business processes and technologies and their proper implementation

It is essential to provide evidence to demonstrate an organization's continued compliance with legislation, policies, principles, processes, and procedures over time.

The principles of good practice in record keeping are of value even if the need to produce electronic records in court never arises. The effort and resource required to comply quickly bring business benefits, whether the organization is in court or not.

Evidential weight

Organizations need to be aware of the potential for legal challenge when documents are presented as evidence in a court of law. If the integrity or authenticity of a record is called into doubt by suggestions of tampering, incompetence, improper system functionality or malfunction, the evidential weight or value put on that document can be lost, or at the very least, reduced, to the detriment of the case.

Therefore, there is a requirement to have readily available evidence to demonstrate and prove the organization's compliance with legislation, policies and procedures throughout the system. You must also show that the system was operating as intended in accordance with the organization's normal business practices. The evidence is available from records of the auditing of system processes.

8.2 Auditing an IBM FileNet Records Manager system

In order to allow organizations to implement a records management system that adheres to best practice principles as outlined in the ISO 15489 standard, a robust framework to capture audit events is a fundamental requirement. Such a framework is provided by the IBM FileNet P8 Platform architecture, and the framework is further augmented when IBM FileNet Records Manager is installed and an object store is enabled to support a records management file plan.

This section does not go into detail about the basic auditing framework provided by IBM FileNet P8 Platform, but it examines auditing for IBM FileNet Records Manager systems.

When a Records Manager data model is imported into an object store, the standard Content Engine (CE) audit events are augmented with the RMAudit events, which are automatically subscribed to for the following classes:

- Record Category
- Record Folder
- Volume

The RMAudit events can be manually configured for Record classes.

The RMAudit event records audit entries whenever any of the following actions are performed on an entity:

- Delete
- Relocate
- Destroy
- Transfer
- Interim Transfer
- Export
- Review when in a Disposition phase

Within **ecm_help** → **Expansion Products** → **Records Manager** → **Auditing**, details are provided of the system events that normally have to be enabled on the relevant object classes within the File Plan Object Store (FPOS).

Where electronic records are being managed, consideration needs to be given to the auditing requirements for the records contained within the FPOS.

By default, auditing is not enabled for an object store; therefore, in order to capture audit events (whether system or RMAudit), auditing on the object store must be enabled.

Details of how to enable auditing on an object store can be obtained by selecting **ecm_help** → **FileNet P8 Administration** → **Content Engine Administration** → **Auditing**.

> **Note:** A best practice for auditing is to only enable the events that require auditing on the classes that need be audited. Do not turn on auditing for all events on the root classes and inherit it to the children. This action will cause a phenomenal amount of audit data to be generated, which will severely impact the performance of running audit reports; the data that is eventually returned to the user will contain superfluous information through which the user must read in order to get to the pertinent information. In addition, it will provide administrative overhead, because there are no predefined tools to manage (export/delete) the audit entries in the databases.

8.2.1 Audit configuration

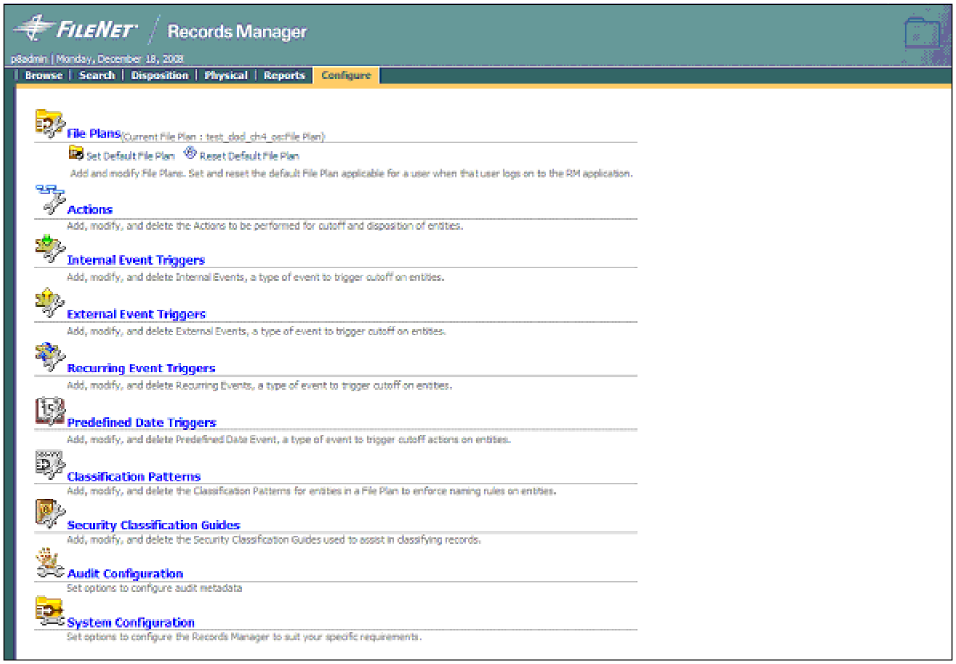
This audit design is based on the principle of not duplicating any functions and features that IBM FileNet Enterprise Manager has already provided. So, from IBM FileNet Enterprise Manager, you select which objects and actions to audit. From IBM FileNet Records Manager, you select the metadata for the audit-enabled objects.

To configure audit metadata, follow these basic steps:

1. Select the audit object and actions from IBM FileNet Enterprise Manager.
2. Select the metadata for each of selected objects from IBM FileNet Records Manager Audit Configuration page.
3. Click **Apply** to complete the task.

To access the Audit Configuration page from the IBM FileNet Records Manager home page:

1. Select **Configure** → **Audit Configuration**. Refer to Figure 8-1 on page 203.



File Plans(Current File Plan : test_dod_ch4_os:File Plan)

Set Default File Plan Reset Default File Plan

Add and modify File Plans. Set and reset the default File Plan applicable for a user when that user logs on to the RM application.

Actions

Add, modify, and delete the Actions to be performed for cutoff and disposition of entities.

Internal Event Triggers

Add, modify, and delete Internal Events, a type of event to trigger cutoff on entities.

External Event Triggers

Add, modify, and delete External Events, a type of event to trigger cutoff on entities.

Recurring Event Triggers

Add, modify, and delete Recurring Events, a type of event to trigger cutoff on entities.

Predefined Date Triggers

Add, modify, and delete Predefined Date Event, a type of event to trigger cutoff actions on entities.

Classification Patterns

Add, modify, and delete the Classification Patterns for entities in a File Plan to enforce naming rules on entities.

Security Classification Guides

Add, modify, and delete the Security Classification Guides used to assist in classifying records.

Audit Configuration

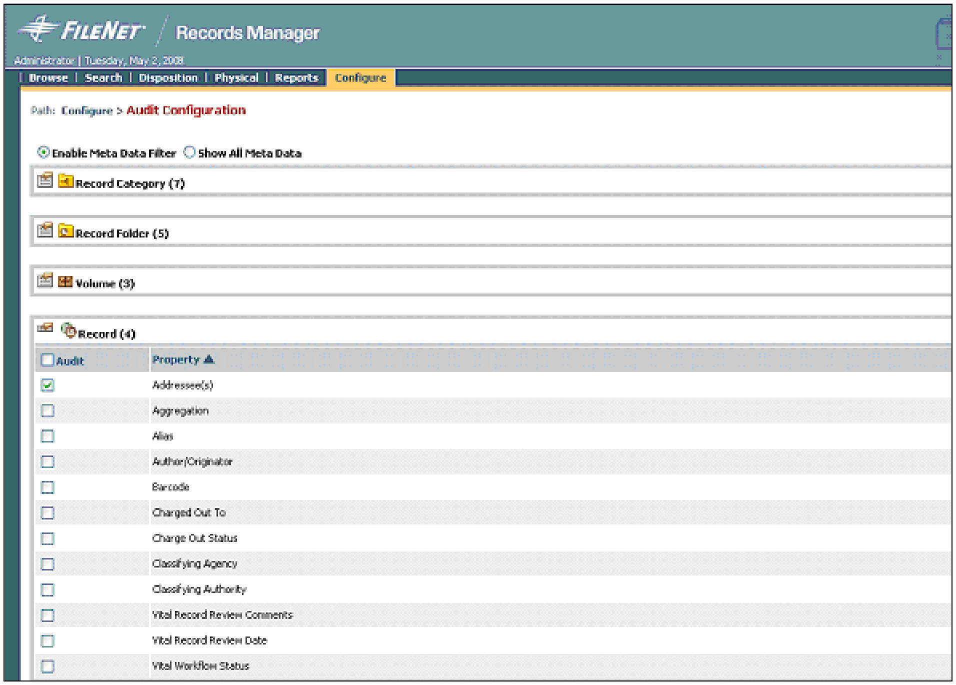
Set options to configure audit metadata

System Configuration

Set options to configure the Records Manager to suit your specific requirements.

Figure 8-1 Accessing Audit Configuration

2. Select the metadata to audit. Refer to Figure 8-2 on page 204.


FileNet
Records Manager
Administrator | Tuesday, May 2, 2006
Browse | Search | Disposition | Physical | Reports | Configure

Path: Configure > Audit Configuration

● Enable Meta Data Filter ○ Show All Meta Data

Record Category (7)

Record Folder (5)

Volume (3)

Record (4)

□ Audit	Property ▲
☑	Addressee(s)
□	Aggregation
□	Alias
□	Author/Originator
□	Barcode
□	Charged Out To
□	Charge Out Status
□	Classifying Agency
□	Classifying Authority
□	Vital Record Review Comments
□	Vital Record Review Date
□	Vital Workflow Status

Figure 8-2 Select metadata to audit

8.2.2 Accessing the audit log

When auditing is enabled on an object store, the audit log entries are stored in a table in the object store database. The audit log stores the following information:

- The action that was performed
- The entity on which the action was performed
- The user who performed the action
- The date and time of the action
- Whether the action succeeded or failed
- For RMAudit events, the reason for performing the action

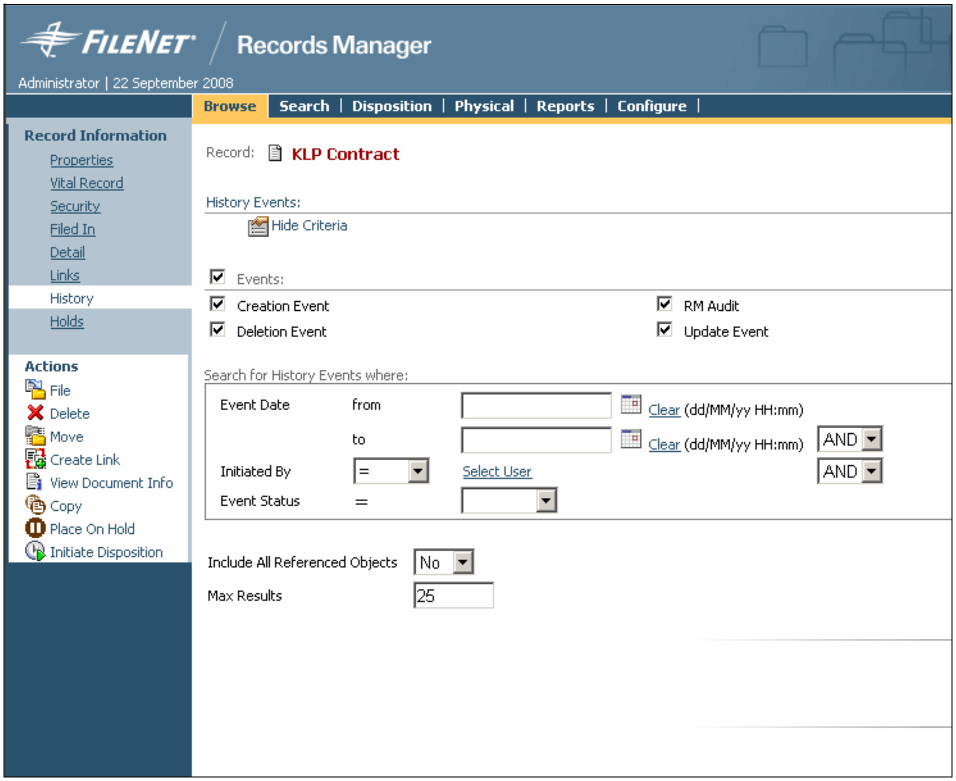
Accessing these entries can either be done on an entity-by-entity basis through the IBM FileNet Records Manager application or searched for in bulk through the query builder in IBM FileNet Enterprise Manager.

Accessing an entity's audit entries

To access an entity's audit entries in IBM FileNet Records Manager systems:

1. Browse through the appropriate folder to find the record, or go to the search page to find the record.

2. After the record has been identified, go to the Record Information and select **History**. Refer to Figure 8-3.



Record: KLP Contract

History Events:

☑ Events:

☑ Creation Event ☑ RM Audit

☑ Deletion Event ☑ Update Event

Search for History Events where:

Event Date from [ ] (dd/MM/yy HH:mm)

to [ ] (dd/MM/yy HH:mm) AND

Initiated By = AND

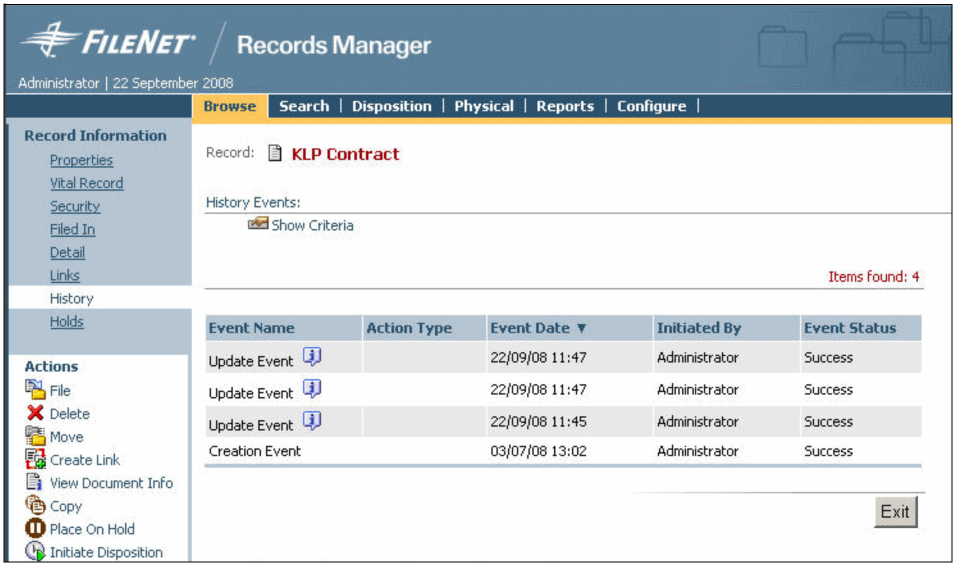
Event Status =

Include All Referenced Objects No

Max Results 25

Figure 8-3 Search audit history page for an entity

3. A search page is displayed that allows you to select which of the enabled audit events to search and provides you with the ability to determine a date range over which to search. After a search is issued, the result is returned (refer to Figure 8-4 on page 206).



Record: KLP Contract

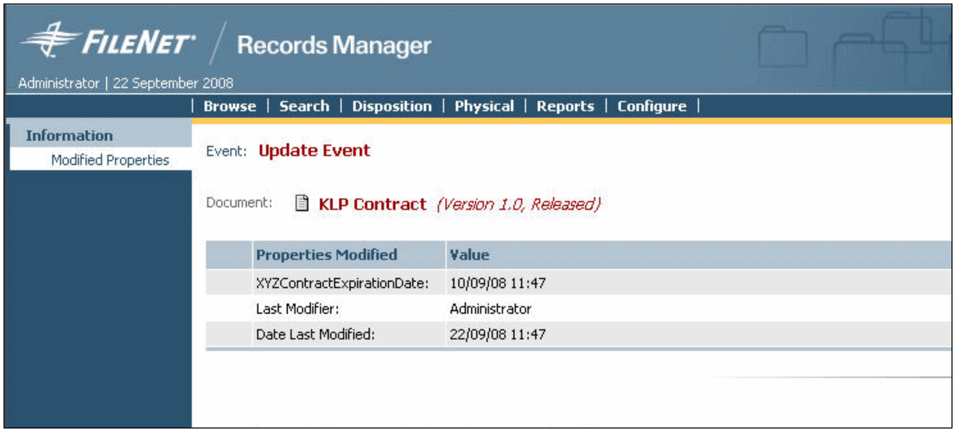
History Events:

Items found: 4

Event Name	Action Type	Event Date ▼	Initiated By	Event Status
Update Event		22/09/08 11:47	Administrator	Success
Update Event		22/09/08 11:47	Administrator	Success
Update Event		22/09/08 11:45	Administrator	Success
Creation Event		03/07/08 13:02	Administrator	Success

Figure 8-4 Audit history results for an entity

4. If an auditable event has further detail associated to it, the information icon displays next to the event name. When you click this icon, the additional information is displayed. In this case, it is the detail of an update event that documents the attributes that were changed on the record and their new values. Refer to Figure 8-5.


FileNet
Records Manager
Administrator | 22 September 2008
| Browse | Search | Disposition | Physical | Reports | Configure |
Information
Modified Properties

Event: Update Event

Document: KLP Contract (Version 1.0, Released)

Properties Modified	Value
XYZContractExpirationDate:	10/09/08 11:47
Last Modifier:	Administrator
Date Last Modified:	22/09/08 11:47

Figure 8-5 Detail of the update audit event

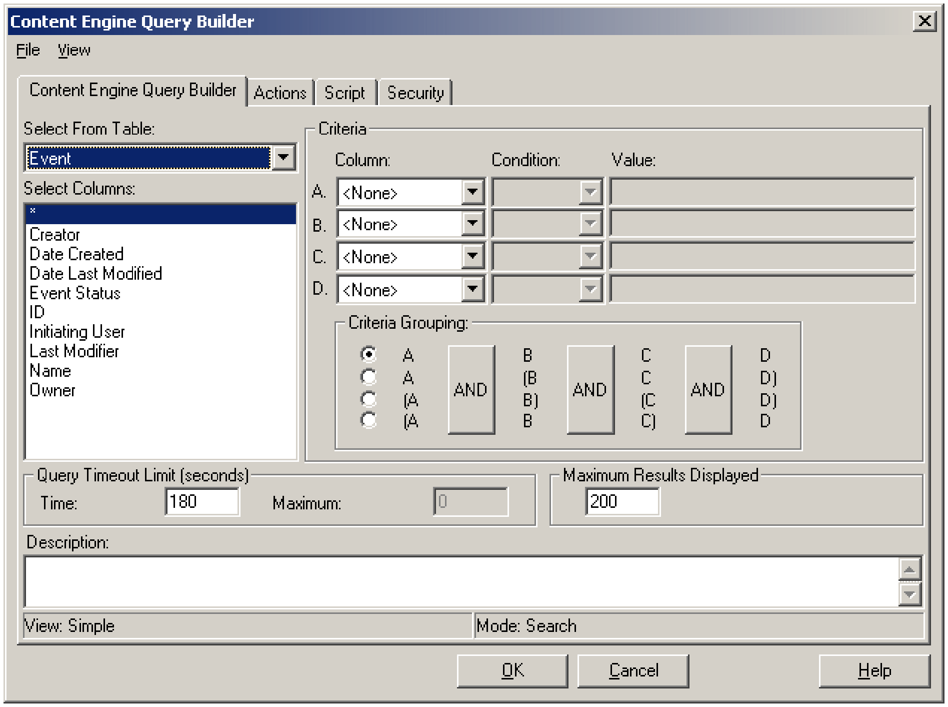
Accessing audit information with FileNet Enterprise Manager

A second method of accessing audit information is through the Query Builder in IBM FileNet Enterprise Manager. Query Builder allows an administrator to get access to all audit information in a single query and, if necessary, to export it to an XML file.

Query Builder allows an administrator to either query for the allowed audit events or to specify a specific event to search. Searchable events include:

- Creation
- Update
- Delete
- RMAudit

To search for all audit events within Query Builder, select the **Event** table (refer to Figure 8-6).



Select From Table: Event

Select Columns:
*
Creator
Date Created
Date Last Modified
Event Status
ID
Initiating User
Last Modifier
Name
Owner

Criteria

	Column:	Condition:	Value:
A.	<None>		
B.	<None>		
C.	<None>		
D.	<None>		

Query Timeout Limit (seconds)
Time: 180 Maximum: 0

Maximum Results Displayed: 200

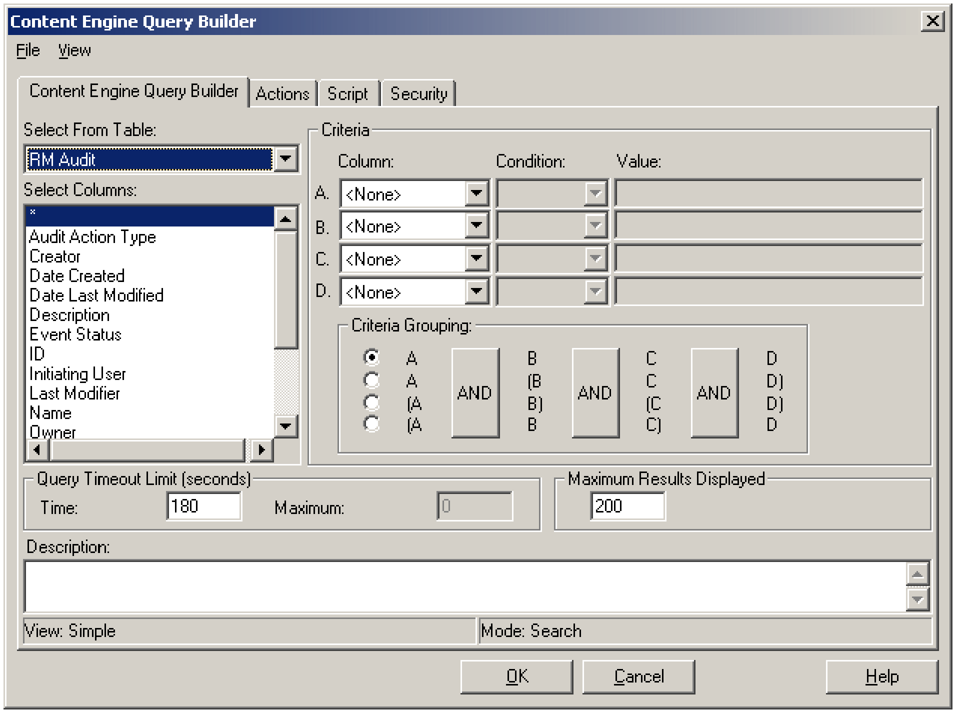
Description:

View: Simple

Mode: Search

Figure 8-6 Query builder searching for Event items

To search for events of a specific type, such as RMAudit, select the relevant event type from the table list (refer to Figure 8-7 on page 208).



Select From Table: RM Audit

Select Columns:

- *
- Audit Action Type
- Creator
- Date Created
- Date Last Modified
- Description
- Event Status
- ID
- Initiating User
- Last Modifier
- Name
- Owner

Criteria

	Column:	Condition:	Value:
A.	<None>		
B.	<None>		
C.	<None>		
D.	<None>		

Criteria Grouping:

● A AND B AND C AND D
○ A AND (B AND C AND D)
○ (A AND B) AND (C AND D)
○ (A AND B AND C) AND D

Query Timeout Limit (seconds) Time: 180 Maximum: 0

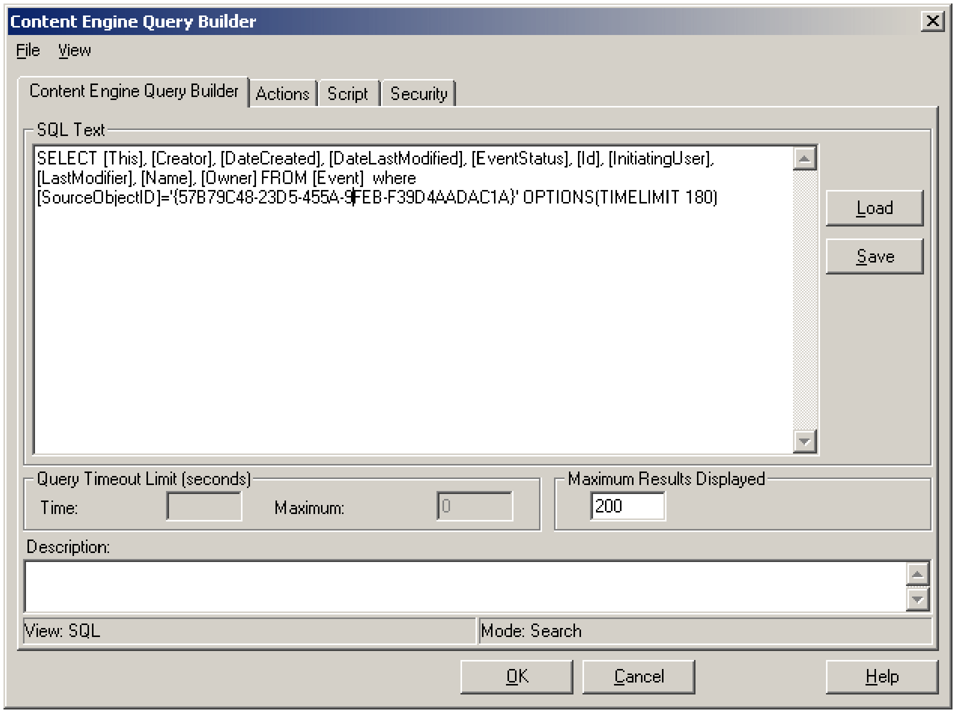
Maximum Results Displayed 200

Description:

View: Simple Mode: Search

Figure 8-7 Query Builder for RMAudit events

If there is a requirement to query for all events for a specific entity, select **Events** from the Table drop-down list, switch to the SQL View for Query Builder, and append a where clause that searches on a specific Object ID (refer to Figure 8-8 on page 209).



```
SELECT [This], [Creator], [DateCreated], [DateLastModified], [EventStatus], [Id], [InitiatingUser],
[LastModifier], [Name], [Owner] FROM [Event]  where
[SourceObjectID]='{57B79C48-23D5-455A-9FEB-F39D4AADAC1A}' OPTIONS(TIMELIMIT 180)
```

Query Timeout Limit (seconds)
Time:
Maximum: 0

Maximum Results Displayed: 200

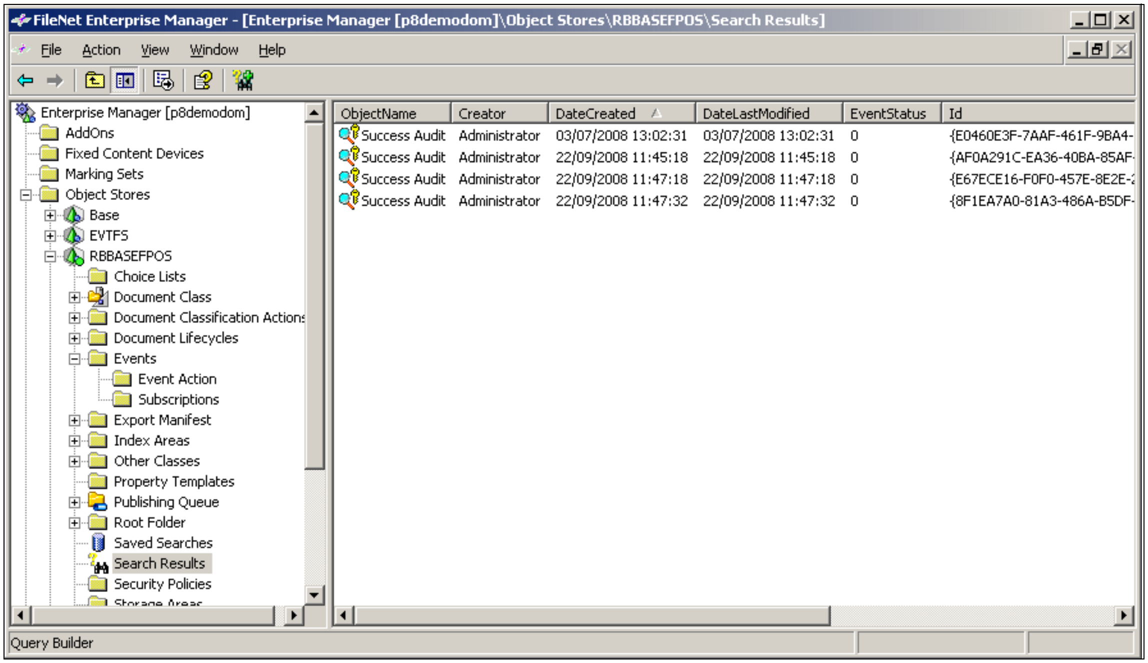
Description:

View: SQL
Mode: Search

Figure 8-8 Using the SQL query view to search for all audit records for a specific entity

Under the Actions tab of the query builder, you can automatically transfer the returned result set (refer to Figure 8-9 on page 210) to the Export manifest for exporting.

Note: All audit information can be exported into an XML file.



ObjectName	Creator	DateCreated	DateLastModified	EventStatus	Id
Success Audit	Administrator	03/07/2008 13:02:31	03/07/2008 13:02:31	0	{E0460E3F-7AAF-461F-9BA4-
Success Audit	Administrator	22/09/2008 11:45:18	22/09/2008 11:45:18	0	{AF0A291C-EA36-40BA-85AF-
Success Audit	Administrator	22/09/2008 11:47:18	22/09/2008 11:47:18	0	{E67ECE16-F0F0-457E-8E2E-
Success Audit	Administrator	22/09/2008 11:47:32	22/09/2008 11:47:32	0	{8F1EA7A0-81A3-486A-B5DF-

Figure 8-9 Audit query results

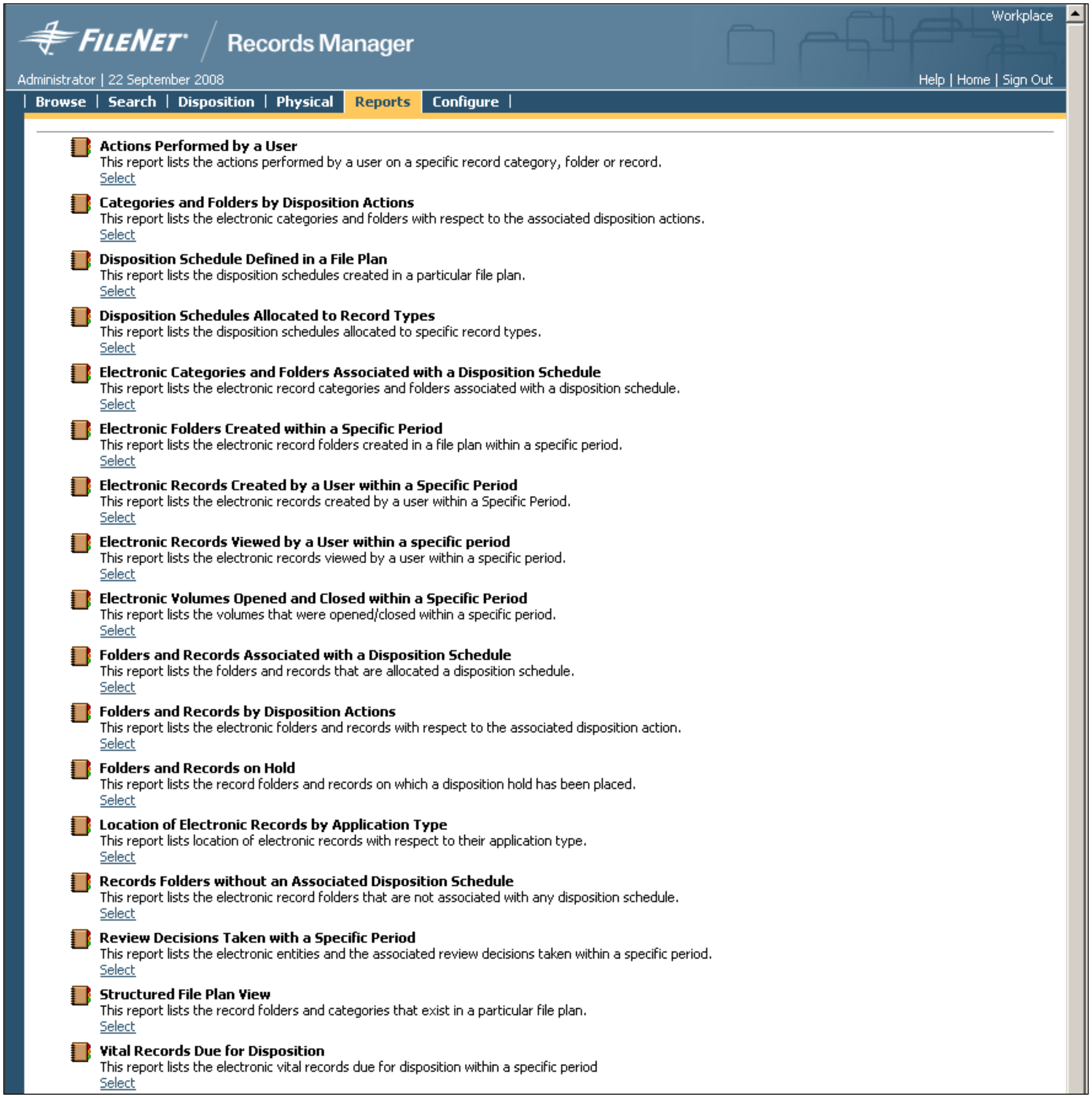
8.3 Reporting

As well as the mechanisms outlined in the previous section for accessing audit information, IBM FileNet Records Manager provides a predefined integration with Crystal Reports that provides a statistical view of activities performed using IBM FileNet Records Manager. An example is a report to show the electronic folders created within a given time period or the review decisions made for entities during a given time period. In addition to using the preconfigured reports, you can also create custom reports.

The interface that Crystal Reports uses to access the data within the IBM FileNet databases is a standard documented interface, which is accessed via JDBC™. Therefore, you can instead use any reporting application that can access data through this type of an interface.

8.3.1 Predefined reports

The IBM FileNet Records Manager application provides 17 predefined Crystal Reports templates that can be accessed through the Reports tab within the IBM FileNet Records Manager application. Refer to Figure 8-10.


FileNet
Records Manager
Workplace
Administrator | 22 September 2008
Help | Home | Sign Out
| Browse | Search | Disposition | Physical | Reports | Configure |

Actions Performed by a User
This report lists the actions performed by a user on a specific record category, folder or record.
Select

Categories and Folders by Disposition Actions
This report lists the electronic categories and folders with respect to the associated disposition actions.
Select

Disposition Schedule Defined in a File Plan
This report lists the disposition schedules created in a particular file plan.
Select

Disposition Schedules Allocated to Record Types
This report lists the disposition schedules allocated to specific record types.
Select

Electronic Categories and Folders Associated with a Disposition Schedule
This report lists the electronic record categories and folders associated with a disposition schedule.
Select

Electronic Folders Created within a Specific Period
This report lists the electronic record folders created in a file plan within a specific period.
Select

Electronic Records Created by a User within a Specific Period
This report lists the electronic records created by a user within a Specific Period.
Select

Electronic Records Viewed by a User within a specific period
This report lists the electronic records viewed by a user within a specific period.
Select

Electronic Volumes Opened and Closed within a Specific Period
This report lists the volumes that were opened/closed within a specific period.
Select

Folders and Records Associated with a Disposition Schedule
This report lists the folders and records that are allocated a disposition schedule.
Select

Folders and Records by Disposition Actions
This report lists the electronic folders and records with respect to the associated disposition action.
Select

Folders and Records on Hold
This report lists the record folders and records on which a disposition hold has been placed.
Select

Location of Electronic Records by Application Type
This report lists location of electronic records with respect to their application type.
Select

Records Folders without an Associated Disposition Schedule
This report lists the electronic record folders that are not associated with any disposition schedule.
Select

Review Decisions Taken with a Specific Period
This report lists the electronic entities and the associated review decisions taken within a specific period.
Select

Structured File Plan View
This report lists the record folders and categories that exist in a particular file plan.
Select

Vital Records Due for Disposition
This report lists the electronic vital records due for disposition within a specific period
Select

Figure 8-10 Predefined report templates

Note: These reports require a full implementation of Crystal Reports Server as documented in the IBM FileNet Records Manager installation and upgrade manual.

Full details of each of these reports and their parameters can be found in the ecm help: **ecm_help** → **Expansion Products** → **Records Manager** → **Report Generation** → **How to...** → **Report Generation and prerequisites**.

These reports can also be customized to an organization’s requirements or if necessary new ones created. Details on how to customize or create your own reports can be found at **ecm_help** → **Expansion Products** → **Records Manager** → **Report Generation**.

8.4 Performance implications

Depending on the size of the IBM FileNet Records Manager system and the amount of audit data that has been captured, generating reports can potentially create a tremendous performance load on the underlying databases. Therefore, for a large system, it is critical to ensure that when reports are generated, the impact on the production environment is minimized as much as possible. For instance, the broader the search criteria, the greater the impact on the performance. All the predefined reports give you a warning if, when requested, a section of the file plan is not specified, (the default is the entire file plan). However, there are other criteria that can also contribute to slow performance. For example, the report, “Electronic Records Created by a User within a Specific Period” allows a date range and one or more users to be entered. Entering long time periods and many user names could result in poor performance.

Audit data can be accumulated over time. If you have auditing enabled, over time, a large quantity of audit data will be gathered. We recommend export the audit information from the system periodically to reduce the load on the system.

9

IBM FileNet Records Manager Java APIs

In this chapter, we describe the IBM FileNet Records Manager Application Programming Interfaces (APIs) with examples.

We describe the following topics in this chapter:

- Introduction to IBM FileNet Records Manager API
- Basic objects
- Records management with the IBM FileNet RM API
- Bulk operations
- Third-party application integration
- Getting a Content Engine session using BCL
- Bulk Declaration Service (BDS)
- Performance considerations

9.1 Introduction to IBM FileNet Records Manager API

The IBM FileNet Records Manager API (RM API) is a set of platform independent application programming interfaces used to interact and perform records management operations on a Content Engine (CE). RM API leverages Version 3.5 of the Content Java API (CE Backward Compatibility Layer (BCL)) for Content Engine operations.

The RM API provides networked, Java-based access to commonly used objects and includes methods for performing record-related operations, such as record declaration, file plan navigation, and record disposition. The RM API provides extensive validation to insure that records manager-related logic is followed for record, file plan, and other records manager-related actions.

You use IBM FileNet Records Manager Java API to:

- Customize an application that uses record management functionality
- Develop new functionality related to records management

For detailed descriptions of the IBM FileNet Records Manager Java API and related methods, refer to IBM FileNet Records Manager documentation.

9.1.1 IBM FileNet Records Manager API requirements

The IBM FileNet Records Manager Java API requires the Java 2 Standard Edition (J2SE™) 1.4.2 or higher development environment. When using the RM API, the following .jar files must be included in your class path:

- `javaapi.jar`
- `pe.jar`
- `rmapi.jar`
- `rmapiresources.jar`
- `rm-bds.jar`
- `accessrole.jar`
- `Jace.jar`

Refer to 15.1, “API development environment setup” on page 390 for more information about setting up the environment.

9.1.2 IBM FileNet Records Manager API class hierarchy

The IBM FileNet Records Manager API leverages the 3.5 version of the CE Content Java API. Most RM API classes are extended CE BCL API classes.

Figure 9-1 illustrates the IBM FileNet Records Manager API class hierarchy. RMObject, RMObjectStore, and RMRecord classes are the three top-level classes in the IBM FileNet Records Manager Java API. All of these classes extend from CE BCL API classes, BaseObject, ObjectStore, and Document. RMFolder also extends from the CE class Folder.

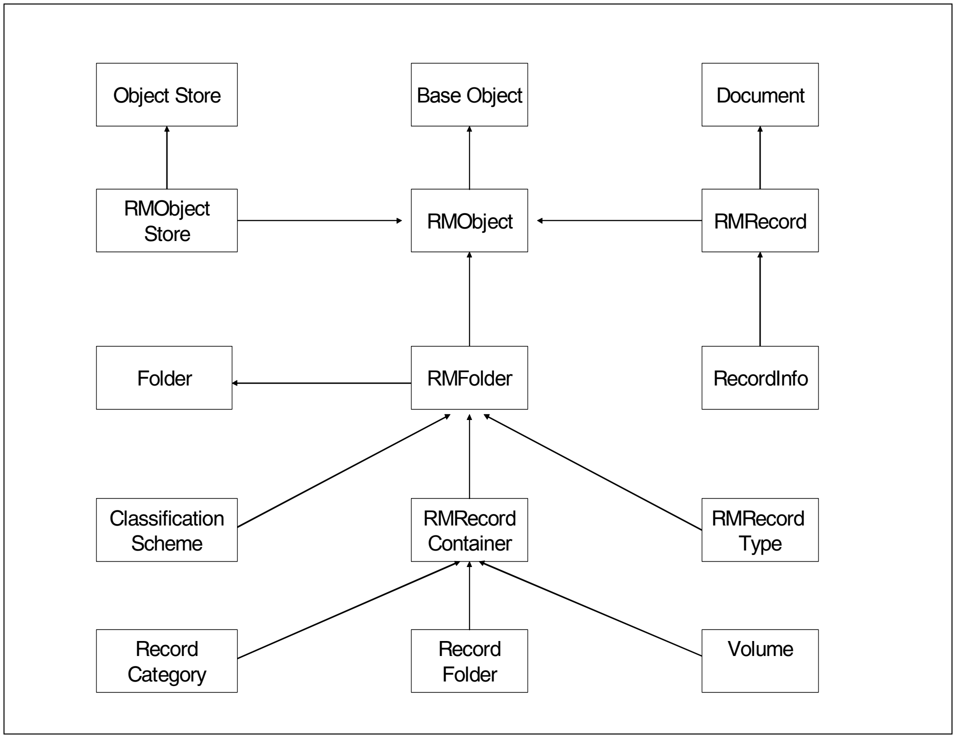


Figure 9-1 IBM FileNet Records Manager API class hierarchy

RMRecordContainer is the container base class for records containers. All container classes, including RecordCategory, RecordFolder, and Volume extend from the RMRecordContainer class. The RMRecordContainer class extends from the RMFolder class, which itself extends from the CE BCL API class and Folder. A record folder is an extension of a document folder. We can say that a document folder with record-specific properties and methods is a record container.

ClassificationScheme and RMRecordType also extend from the RMFolder class.

9.2 Basic objects

When using the IBM FileNet Records Manager APIs, the basic objects we work with include the Content Engine high-level objects and the IBM FileNet Records Manager API base objects.

9.2.1 Content Engine high-level objects

The Content Engine (CE) high-level objects provide access to object store content persistent in a Content Engine.

CE high-level objects are:

- Session
- Object store
- Entire network
- Domains
- Realm

Session

A *session* object is required to establish a Content Engine server connection. A Session object is required to perform any operation on Content Engine.

A session object can be instantiated as:

```
Session ceSession=ObjectFactory.getSession(<app ID>, <encryption 
level>, <user name>, <password>)
```

Where:

- *ObjectFactory* is the base object class, responsible for building the core framework objects.
- <*app ID*> can be any arbitrary name. This name is used to maintain a session with Content Engine.
- <*encryption level*> can be either clear or symmetric. When the value null is specified, the clear encryption level is used.
- <*user name*> is the Content Engine login user ID.
- <*password*> is the Content Engine password for the specified user.

Object store

An *ObjectStore* object represents the reference to a Content Engine object store.

An ObjectStore object can be instantiated as:

```
ObjectStore ceObjectStore = ObjectFactory.getObjectStore(<object store name>, ceSession)
```

The value *ceSession* is a session object that was created in the previous section.

Entire network

An *Entire network* object is the topmost-level object in the hierarchy. It provides the entire structure of the IBM FileNet P8 domain and object stores.

An entire network object can be instantiated as:

```
EntireNetwork entireNetwork=ObjectFactory.getEntireNetwork(ceSession)
```

Domains

A *Domain* object represents the IBM FileNet P8 domain as defined in the Global Configuration Database (GCD) on the Content Engine server. Mainly, a domain is used for retrieving the object stores, users, and groups available in that domain.

A Domain object can be instantiated as:

```
Domains ceDomains = entireNetwork.getAvailableDomains()
```

Realm

A Realm object represents the collection of users and groups.

A Realm object can be instantiated as:

```
Realm realm = entireNetwork.getUserRealm()
```

To get a collection of all the groups associated with the current realm:

```
Groups groups = realm.getGroups(null)
```

To get a particular group from the groups collection:

```
Group group = groups.get(0)
```

To get a user collection and a particular user:

```
Users users = group.getUsers()
User user = users.get(0)
```

When using collections, such as groups and users, you can pass an index parameter to identify which item in the collection to return. In the previous examples, using an index of zero returns the first item in the collection.

Using a Realm object, we can retrieve security groups and users.

9.2.2 IBM FileNet Records Manager Java API base objects

These objects serve as base objects to provide records management operations. The IBM FileNet Records Manager (RM) Java API leverages the Content Engine Java API. Most of the IBM FileNet Records Manager interfaces extend from the 3.5 version of the Content Engine Java interfaces. Refer to the IBM FileNet Records Manager online help and the *Records Manager Java API Reference* for additional information about these objects.

We discuss the following base objects:

- RMObject
- RMObjectStore
- RMFolder
- RMRecord
- RMCustomObject
- RMSearch

RMObject

The *RMObject* interface is the base interface for all IBM FileNet Records Manager objects, and it extends from the BaseObject and ReadableMetadataObject interfaces of the CE Java API.

An RMObject can be instantiated as:

```
RMObject rmObject =
(RMObject)rmObjectStore.getObject(RMType.RM_TYPE_Object, rmObjectId)
```

RMObjectStore

RMObjectStore represents the object store in which all file plans are created and stored. It extends from the RMObject interface of the IBM FileNet Records Manager Java API and the ObjectStore interface of the CE Java API.

An RMObjectStore object can be instantiated as:

```
RMObjectStore rmObjectStore=RMUtil.getRMObjectStore(ceObjectStore)
```

The `ceObjectStore` is a CE Java API ObjectStore object.

RMFolder

The *RMFolder* interface serves as the base interface for all container objects in the IBM FileNet Records Manager Java API. Container objects can be ClassificationScheme, RecordCategory, RecordFolder, or Volume.

The RMFolder interface extends the RMObject interface of the IBM FileNet Records Manager Java API and the Folder interface of the CE Java API.

An RMFolder object for record category can be instantiated as:

```
RMFolder rmFolder =
(RMFolder)rmObjectStore.getObject(RMType.RM_TYPE_RECORDCATEGORY,
rmObjectId)
```

The *rmObjectId* is the Globally Unique Identifier (GUID) of the record category.

RMRecord

The *RMRecord* object represents a record. It extends the RMObject interface and the CE Java API interface Document.

An RMRecord Object can be instantiated as:

```
RMRecord rmRecord =
(RMRecord)rmObjectStore.getObject(RMType.RM_TYPE_ELECTRONICRECORD,
rmObjectId)
```

RMCustomObject

RMCustomObject is the base object for all IBM FileNet Records Manager custom objects. The RMCustomObject interface extends from the CE Java API interface CustomObject.

An RMCustomObject can be instantiated as:

```
RMCustomObject rmCustObj =
(RMCustomObject)rmObjectStore.getObject(RMType.RM_TYPE_ACTION,
rmObjectId)
```

RMSearch

The *RMSearch* interface defines and implements operations for searching IBM FileNet Records Manager entities. It extends from the CE Java API Search interface.

An RMSearch object can be instantiated as:

```
RMSearch rmSearch = rmObjectStore.getRMSearch()
```

9.2.3 Creating a Records Manager File Plan Object Store object

Example 9-1 shows a sample program to create an object representing the IBM FileNet Records Manager File Plan Object Store (FPOS). The sample has been simplified by leaving out the error-handling code. For production code, you need to add sufficient error handling to insure the proper execution of the program.

This example performs the following tasks:

1. Get a Content Engine session.
2. Get the Content Engine object store object.
3. Get the Records Manager File Plan Object Store (FPOS).
4. Get the name and GUID of the FPOS.

Example 9-1 A sample program to get a Records Manager object store

```
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import com.filenet.rm.api.RMObjectStore;
import com.filenet.rm.api.util.RMUtil;
import com.filenet.wcm.api.ObjectFactory;
import com.filenet.wcm.api.ObjectStore;
import com.filenet.wcm.api.Session;

public class Sample
{
  public static void main( String [] args )
  {
  // Set user name and password for the Content Engine server.
  // We hard-coded user name and passwords.
  // Make sure you replace them (Administrator and filenet)
  // with your Content Engine user name and password.

  String userName = "Administrator";
  String password = "filenet";

  // Set File Plan Object Store name.
  // We hard-coded the object store name.
  // Make sure you replace it (RBBASEFPOS)
  // with your File Plan Object Store name.

  String objectStoreName = "RBBASEFPOS";

  // 1. Get a Content Engine session.
  Session ceSession = ObjectFactory.getSession
```

```
    ("RM", null, userName, password);
  try
    {
    // Specify the WcmApiConfig.properties path according
    // to your FileNet application installation.
    ceSession.setConfiguration(new FileInputStream
      ("C:\\Program Files\\FileNet\\AE\\Workplace\\
      WEB-INF\\WcmApiConfig.properties"));
  }catch(FileNotFoundException fe){}
  System.out.println("Session created");

  // 2. Get the Content Engine object store object.
  ObjectStore ceObjectStore = null;
  ceObjectStore = ObjectFactory.getObjectStore
    (objectStoreName, ceSession);
  System.out.println("Got the CE OS object");

  // 3. get the RM object store (which is the FPOS) object.
  RMUtil rmutility = new RMUtil();
  RMObjectStore loRMObjectStore =
    rmutility._getRMObjectStore(ceObjectStore);
  System.out.println("Got the RM Object Store");

  // 4.Getting the name and GUID of the Object Store.
  String ObjName = loRMObjectStore.getName();
  System.out.println("Object Store Name: "+ ObjName);
  String Id = loRMObjectStore.getId();
  System.out.println("Object Store GUID: "+ Id);
  }
}
```

For our case study environment, the output of the sample code is shown in Example 9-2.

Example 9-2 Sample code output

```
Session created
Got the CE OS object
Got the RM Object Store
Object Store Name: RBBASEFPOS
Object Store GUID: {B8CE383D-EE40-437F-A9B2-F47B4C782D14}
```

You can get GUIDs for objects other than the object store in the same manner. In the next sections, we use these high-level objects to perform records management operations.

9.3 Records management with the IBM FileNet RM API

Records management includes declaring a document as a record and operations, such as move, copy, and file, on records and containers. Most of the records management operations can be performed through the application's user interface. However, by leveraging the IBM FileNet Records Manager (RM) Java API, you can customize these records management operations for your business requirement.

This section illustrates records management operations by using the IBM FileNet Records Manager Java API.

9.3.1 Record declaration

We declare an existing document as a record, so we assume the document is already stored in the IBM FileNet Content Manager repository (in Content Engine). The document must have been created in a Records-enabled Object Store (ROS) in order for it to be declared as a record. We will use the RM API to interact with the CE Java API to create a RecordInfo object that corresponds to the Document object.

During the declaration process, our sample API validates whether the document can be declared as a record. In addition, it validates whether the destination container object can hold the RecordInfo object.

Perform the following steps to declare the document as a record:

1. Get a Content Engine session:

   ```
   Session ceSession = ObjectFactory.getSession(AppId, null, userId, 
   password)
   ```

2. Get the Records Manager object store RMObjectStore:

   ```
   ObjectStore ceObjectStore = 
   ObjectFactory.getObjectStore("ObjectStoreName", session)
   RMObjectStore rmObjectStore=new 
   util().getRMObjectStore(ceObjectStore)
   ```

3. Get the destination container object:

```
RMRecordContainer rmRecordContainer =
(RMRecordContainer)rmObjectStore.getObject(RMType.RM_TYPE_ENTITY,GUI
D)
```

In the *RM_TYPE_ENTITY*, you can specify the container type, and GUID is the ID of the container.

4. Create new property instances:

```
Property loDate =
ObjectFactory.getProperty(RMProperty.DATE_RECEIVED)
Property loMedia = ObjectFactory.getProperty(RMProperty.MEDIATYPE)
Property loFormat = ObjectFactory.getProperty(RMProperty.FORMAT)
Property loPubDate =
ObjectFactory.getProperty(RMProperty.PUBLICATIONDATE)
Property loAuthor = ObjectFactory.getProperty(RMProperty.AUTHOR)
Property loOrgn =
ObjectFactory.getProperty(RMProperty.ORIGINATING_Organization)
Property loSubjTitle = ObjectFactory.getProperty(RMProperty.SUBJECT)
Property lsTitle =
ObjectFactory.getProperty(Property.DOCUMENT_TITLE)
```

5. Set the property values:

```
loDate.setValue(new java.util.Date())
loMedia.setValue("Text")
loFormat.setValue("Text")
loPubDate.setValue(new java.util.Date())
loAuthor.setValue("Author_name")
loOrgn.setValue("origin")
loSubjTitle.setValue("subjtitle")
lsTitle.setValue("title")
```

6. Add all the properties to the properties collection:

```
Properties loProps = ObjectFactory.getProperties()
loProps.add(loDate)
loProps.add(loMedia)
loProps.add(loFormat)
loProps.add(loPubDate)
loProps.add(loAuthor)
loProps.add(loOrgn)
loProps.add(loSubjTitle)
loProps.add(lsTitle)
```

7. Declare and file the new record in the RecordContainer:

```
RecordInfo loRecord = rmRecordContainer.declare (new
RMUtil()._getDocumentStore(rmObjectStore), new String[]{" Doc ID"},
null, loProperties, "RecordClassId", null, false)
```

RecordInfo is the declared record. The declare() method has the following structure:

```
public RecordInfo declare(DocumentStore aoDocumentStore,
java.lang.String[] asDocumentsID, RMFolder aoFolder,
com.filenet.wcm.api.Properties aoRecordProps, java.lang.String
asRecordInfoClassID, com.filenet.wcm.api.Permissions aoRecordACLs,
boolean abAutoname)
```

Where:

- *aoDocumentStore* is the wrapper on the DocumentStore. It stores all the documents to be declared as a record.
- *asDocumentsID* specifies an array of document IDs that are to be declared as a record.
- *aoFolder* is the list of additional record containers (each of which can be either a record category or record folder) where this record needs to be filed during declaration.
- *aoRecordProps* is the properties of the Record.
- *asRecordInfoClassID* is the ID of the record class on which to base the new record, typically ElectronicRecordInfo, Marker, or EmailRecordInfo.
- *aoRecordACLs* is the optional Permissions collection specifying the access permissions for the record. Typically, this parameter is set to null to allow the use of the specified record class' Default Instance Permissions.
- *abAutoname* is the Boolean indicator for autoName of record. If autoName is true, the API automatically generates the record name using the registered AutoName implementation. If autoName is false, the name is taken from the Document Title property.

For a more detailed description of the declare() method, refer to ecm_help.

9.3.2 Basic records management operations

In addition to records declaration, the basic records management operations that you can perform with IBM FileNet Records Manager APIs include:

- Updating a record's metadata
- Copying a record
- Moving a record
- Superceding a record

Updating a record’s metadata

The property values (metadata) of a record can be modified through APIs. Updating a record’s metadata consists of the following steps:

1. Get the record object:

   ```
   RMRecord loRMRecord =
   (RMRecord)loRMObjectStore.getObject(RMType.RM_TYPE_ELECTRONICRECORD,
   lsObjectId)
   ```

   Make sure that you get the lsObjectId first before you actually get the record object.

2. Retrieve the property that you want to update:

   ```
   Properties loProperties = ObjectFactory.getProperties()
   Property loDate = ObjectFactory.getProperty(“PUBLICATIONDATE”)
   ```

   In the first line, getProperties() gets a collection of properties. From the properties collection, you retrieve the property object by providing the symbolic name of the property on the second line. For our example, we update a property `PUBLICATIONDATE`; thus, we pass that name to get the object. Other properties can be updated by using the specific property name.

3. Update the property:

   ```
   loDate.setValue(publicationDate)
   loProperties.add(loDate)
   ```

   The publicationDate is a date object that stores the publication date of the record. We set the loDate (which is the PUBLICATIONDATE property object) with the value of publicationDate.

4. Set the property on the Record object:

   ```
   loRMRecord.setProperties(loProperties)
   ```

The same process can be followed to update a folder and other objects’ properties. For updating a folder, pass `RMType.RM_TYPE_ELECTRONICRECORDFOLDER` instead of `RMType.RM_TYPE_ELECTRONICRECORD` when getting the folder object (refer to step 1).

To update any object’s property, get the object first, then get its property object, set its value, and set the property back to the object.

> **Note:** System properties cannot be updated by users, because the system properties are created by the system and not by the users.

Copying a record

Copying a record involves creating an identical copy of the record. The copy operation is helpful in archiving. It leaves the original record intact and creates a new record copy at the destination directory.

In the copy operation, a new RecordInfo object is created. A copy operation results in two identical copies of the record (RecordInfo) object referencing the same document.

A copy operation is performed using the following code:

```
aoSource.copy(aoDestination, aoDestinationRecProps)
```

Where:

- *aoSource* is the source record. We can instantiate the source record as described in 9.2.2, “IBM FileNet Records Manager Java API base objects” on page 218.
- *aoDestination* is the destination container. A destination container (here a record category) can be instantiated as:

  ```
  RecordCategory loRecCategory = (RecordCategory)
  loRMStore.getObject(RMType.RM_TYPE_RECORDCATEGORY, "RecCatName")
  ```
- *aoDestinationRecProps* is a collection of properties for the new record. It cannot be null, because at least the title of the new record must be provided in the properties collection.

Moving a record

Moving a record means relocating the record to a new container.

This operation is used for records relocation. For a sample scenario, certain records are declared first in the Human Resource container. At a later point in time, the records administrator realizes that these records need to be in a new container called Facility Management. In this scenario, these records need to be moved.

In the move operation, a source record is removed from its original container and placed in the new container. After the move, the record will be assigned the schedule that is associated with its new location in the file plan.

The move operation is performed using the following code:

```
loRecordInfo.move (loSourceContainer, loDestinationContainer,
lsReasonToMove)
```

Where:

- *loRecordInfo* is the source record.
- *loSourceContainer* is the source record's container.
- *loDestinationContainer* is the destination record's container.
- *lsReasonToMove* is a string that specifies the reason to move the record.

Superceding a record

In a supercede operation, one record (superceding record) overrides another record (superceded record). Superceding is performed when an older record is no longer in use and needs to be replaced by a new record.

Supercede is performed using the following code:

```
aoSupercedingRecord.supercede(aoSupercededRecord)
```

Where

- *aoSupercedingRecord* is the record that is superceding.
- *aoSupercededRecord* is the record being superceded.

> **Note:** *Superseding* is extremely useful when working with a versioned document. If an older version of a document has been declared as a record, and a new version of the document has also been declared as a record, they are separate records that do not relate to each other. If your business requires that the older record must be superceded with the newer record, you use the supercede() method. IBM FileNet Records Manager does not perform supercede automatically.

For a more detailed description of these methods, refer to ecm_help.

9.3.3 Container object operations

Containers can be categories, folders, or volumes. In this section, we discuss operations on these containers through the use of the IBM FileNet Records Manager Java API:

- Closing and reopening a container object
- Activating and deactivating a container object

Closing and reopening a container object

If a container is closed, you cannot add any more records or children to that container. To continue adding records or children to a container, you need to reopen it.

A container can be closed using the following code:

```
aoObj.close(asReasonForClose)
```

Where

- *aoObj* is the container object that needs to be closed.
- *asReasonForClose* is a string that specifies the reason to close the container.

A container can be reopened using the following code:

```
aoObj.reOpen(abReopen,asReasonForClose)
```

Where:

- *aoObject* is the container object that needs to be reopened.
- *abReopen* is a Boolean argument that sets the reopen.
- *asReasonForOpen* is the reason to reopen the container.

Note: If the *abReopen* argument is set to false, the *reOpen* method closes the container. In this case, the *reOpen* method behaves as the *close* method.

Activating and deactivating a container object

No operation can be performed on an inactive container. A container can be deactivated using the following code:

```
aoObj.inActivate(abInactive, asReasonForInactivate)
```

Where:

- *aoObj* is the container object that needs to be deactivated.
- *abInactive* is the flag that needs to be set to true to deactivate the object.
- *asReasonForInactivate* is the reason for deactivating the container.

If the flag *abInactive* is set to false, the inActivate method activates the container.

9.3.4 Bulk operations

A *bulk operation* is used to perform an action on multiple objects. The RM API provides bulk operation support. Several of the common bulk operations are:

- Closing and reopening multiple containers
- Activating and deactivating multiple records
- Holding, removing holds, and relocating multiple records

Closing a container

Using the close bulk operation, multiple containers can be closed in a single request, which improves the performance when running this operation on multiple entities.

The close bulk operation is performed using the following code:

```
RMBulkOperationResults loResults = new RMBulkOperationsUtil().close
(aoRMOS, lsArrIDs, "ReasonToClose")
```

lsArrIDs is an array of strings that contains the GUID of the containers to close.

> **Note:** Do not confuse the bulk operations provided by IBM FileNet Records Manager APIs with Bulk Declaration Services (BDS). We discuss BDS in 9.6, “Bulk Declaration Service (BDS)” on page 232.

Reopening a container

The reopen bulk operation opens multiple closed containers in one request.

The reopen bulk operation is performed using the following code:

```
RMBulkOperationResults loResults = new RMBulkOperationsUtil().reOpen
(aoRMOS, lsArrIDs, true, "test")
```

lsArrIDs is an array of strings that contains the GUID of the closed containers that need to be opened.

For more detailed information about bulk operations, refer to ecm_help.

9.4 Third-party application integration

Many times, there are situations when a document is generated and stored in a system other than IBM FileNet P8 content repository. Users might need to declare records directly from that system. In this case, IBM FileNet Records Manager needs to communicate with that system.

IBM FileNet Content Federation Services federates content from heterogeneous repositories into the IBM FileNet P8 content repository. With IBM FileNet Content Federation Services, you can put metadata from disparate content repositories into the IBM FileNet P8 Content Engine's master catalog, thereby making that metadata available. IBM FileNet Content Federation Services can enforce records management policies across repositories and in response to business events automatically.

For more information about IBM FileNet Content Federation Services, refer to:

http://www.ibm.com/software/data/content-management/filenet-content-manager/federation.html?S_CMP=rnav

9.5 Getting a Content Engine session using BCL

Records Manager Application uses Content Engine (CE) sessions to access CE objects. As shown in Figure 9-2 on page 231, RM uses *Backward Compatibility Layer (BCL)* to communicate with the CE server. BCL can interact with the CE server using either an Enterprise Java Bean (EJB™) connection or a Content Engine Web Services (CEWS) protocol.

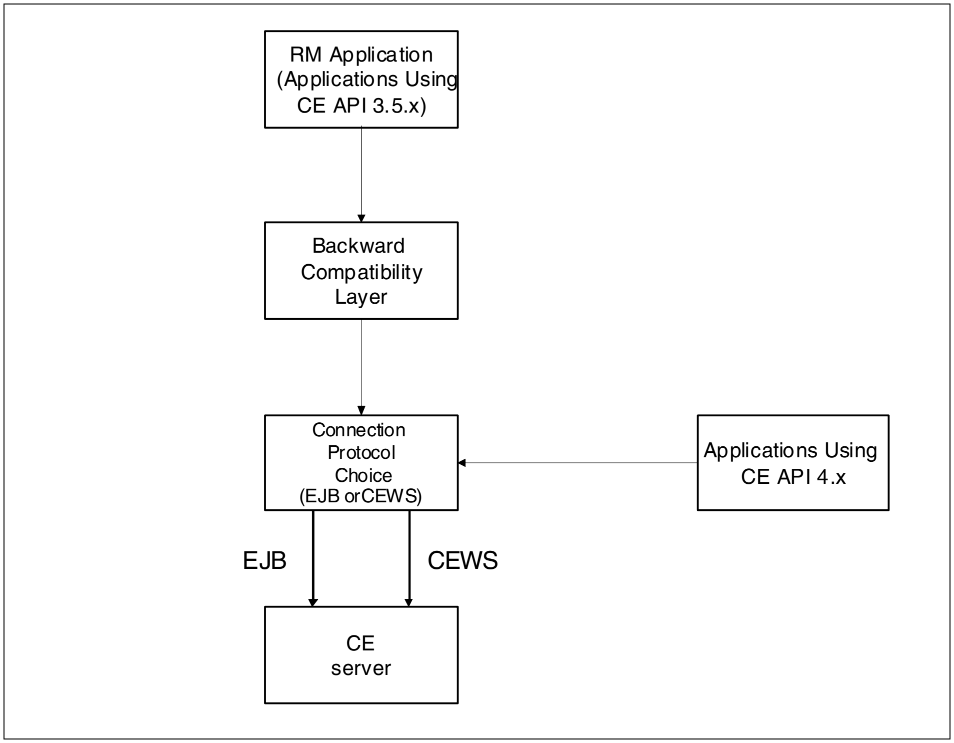


Figure 9-2 Communication with CE server

BCL provides an interface to the CE Java API 3.5.*x* to communicate with Content Engine 4.*x*. IBM FileNet Records Manager uses CE Java API 3.5.*x* and hence communicates with CE using the BCL. BCL is provided to support older applications. Applications using CE Java API 4.*x* can interact with the CE without any need of the BCL. CE Java API communicates with Content Engine using one of the following two protocols:

- EJB: The EJB protocol passes serialized Java objects across the network for requests and responses to and from the Content Engine server. It fully utilizes the Java 2 Platform, Enterprise Edition (J2EE) framework capabilities and uses Java Authentication and Authorization Service (JAAS) for authentication and single sign-on (SSO). EJB is the fastest transport interface available for CE 4.*x*.
- CEWS: CEWS is also referred to as Web Services Interoperability (WSI). This protocol is based on the Simple Object Access Protocol (SOAP).

  In the CEWS protocol, Content Engine objects are represented in serialized XML form and passed across the network from the client to the Content Engine server.

Getting a CE session using the EJB transport layer is complex and out of the scope of this book. Here, we demonstrate how to get a CE session using the CEWS interface.

To get a CE session, follow these steps:

1. Set up a WcmApiConfig.properties file:

   Update WcmApiConfig.properties file for the following entries:

   ```
   RemoteServerUrl = cemp:http://<server>:9080/wsi/FNCEWS40DIME
   RemoteServerUploadUrl = cemp:http://<server>:9080/wsi/FNCEWS40DIME
   RemoteServerDownloadUrl = cemp:http://<server>:9080/wsi/FNCEWS40DIME
   ```

   Replace <*server*> with your server name or server IP address. These entries are specific to IBM WebSphere Application Server 6.*x*.

2. Use this code to get a CE session:

   ```
   Session ceSession = ObjectFactory.getSession(AppId, null, userName, 
   password)
   ```

3. Specify the WcmApiConfig.properties path according to your FileNet application installation directory:

   ```
    ceSession.setConfiguration(new FileInputStream
      ("C:\\Program Files\\FileNet\\AE\\Workplace\\
      WEB-INF\\WcmApiConfig.properties"))
   ```

In this manner, your RMA can communicate with the Content Engine using the CEWS protocol.

9.6 Bulk Declaration Service (BDS)

Bulk Declaration Service (BDS) is a mechanism to perform multiple record operations in batches. BDS is implemented using a set of interfaces and classes known as the BDS API.

Using BDS, you can perform the following operations:

- Bulk declaration of electronic records from existing documents in CE
- Bulk declaration of physical records
- Bulk creation of new CE documents and declaring them as records
- Bulk creation of new CE documents

Most BDS operations can be performed using IBM FileNet Records Manager APIs as well, but BDS is much faster due to the batch operation.

Using BDS creates overhead. If you have only a few records to declare at a time, we recommend using IBM FileNet Records Manager APIs. If you have to declare a high volume of records, we recommend using BDS.

The BDS API consists of the following packages:

- *com.filenet.rm.bds*

  This package contains most of the BDS interfaces and classes.
- *com.filenet.rm.bds.exception*

  This package contains BDS exception implementation classes.
- *com.filenet.rm.bds.impl*

  This package contains the implementation classes of the BDS interfaces.

BDS works in the following manner:

1. Create a new batch.
2. Add record declaration or document creation requests to the newly created batch.
3. Request the execution of the batch.

The power of BDS lies in batching, which results in a savings of overall execution request time.

BDS is extremely useful when a new records management environment is being set up and thousands of documents need to be declared as records. A custom stand-alone application can be developed using the BDS API to declare records quickly.

For BDS interfaces and classes detail, refer to ecm_help.

9.7 Performance considerations

The following suggestions result in performance improvement:

- While querying Content Engine for data, retrieve only the minimum data required to complete the task. Minimize data retrieval in the following ways:
  - If executing a CE SQL statement, select only the necessary columns.
  - If executing a getObject() request, use a property filter and only request the necessary properties.
  - Use paging when the resultset can be large. Returning a large resultset can consume too much memory.

  - If possible, process the resultset immediately and do not save the data for later use, which ensures that memory consumption is kept to a minimum.
  - If serializing and deserializing data, use light binary objects, such as vectors and maps, or use the file system. Avoid serializing and deserializing information to XML unless the information is directly consumed by a client application that expects XML.
- Batch the write operations if possible. For example, when creating multiple CE instances (such as documents or custom objects), updating multiple CE properties, or deleting multiple CE instances, batch those requests together. For optimum performance, you need to limit the batching according to the resources available. So, a balanced batching results in high performance.

10

System to system transfer

In this chapter, we describe the IBM FileNet Records Manager System to System Transfer capability. This capability allows a records manager to move records folders and records from one File Plan Object Store (FPOS) to another.

In this chapter, we describe the following topics:

- System to system transfer overview
- System to system transfer functionality
- Creating import and export mappings
- Exporting records and records folders
- Importing records and records folders

10.1 Overview of system to system transfer

The Records Manager Transfer tool provides the capability to move records and records folders from one Records Management Application (RMA) to another RMA. This capability includes moving items from one File Plan Object Store (FPOS) to another FPOS. When a record or folder is moved, the content, as well as the metadata, is also moved. The Records Manager Transfer tool only moves records and records folders. To move Categories, use the Records Manager File Plan Import Export tool.

The Records Manager Transfer tool provides both an import and an export facility. The export generates a set of XML files for the records and records folders that are exported. The XML format is based on a schema provided by the Joint Interoperability Test Center (JITC). JITC is responsible for performing certification tests for RMAs seeking certification based on the Department of Defense (DoD) 5015.2 standard. The new Version 3 of this standard has added the system to system transfer capability.

IBM FileNet Records Manager supports both physical and electronic records, and both types of records can be exported and imported. For electronic records, the content (document) is included in the generated XML. The content is encoded before it is added to the XML to insure that the XML file is well-formed. Because physical records do not contain content elements, the XML for content is not be created when exporting physical records.

IBM FileNet Records Manager supports declaring documents with multiple content elements as a record. In addition, a record can incorporate multiple documents or multiple versions of the same document. In all these cases, all of the content objects are added to the XML file in separate nodes so that they can be extracted when the record is imported.

The new export capability provided for system to system transfer can also be used for normal disposition-based transfers. These transfers occur as a part of disposition and are completed through a workflow process. New workflows have been created for Export, Transfer, and Interim transfer that use the new export format and schema and are customized through the use of transfer mappings. In addition, the legacy workflows using the previous export format are still supported and included in the product.

10.1.1 Export

When performing an export, the Records Manager can modify the fields that are exported and, thus the associated schema, by using the RM mapping tool to identify additional fields that must be exported and optionally renaming non-mandatory exported fields. This mapping capability allows the record manager to customize the data that is exported as well as the format of the exported data.

At export time, the user specifies which records to include in the export by directly specifying a list of records to export, or the user can specify a container, in which case, all records in that container are exported. During the export, all records folders associated with the exported records are also automatically exported. In addition, information about the associated folders is included in the record XML.

10.1.2 Import

There is also a mapping functionality for the import functionality. The import mapping allows the records manager to map fields contained in the import schema to the correct fields in the target repository. For instance, if the import schema contains a string field named Subject, the records manager can map this to a record metadata field called Title. This capability allows the import to properly map the input data defined in the schema to the record metadata fields defined in the repository.

When importing a record, the associated content is extracted from the XML and a document is created on the Record Object Store specified in the import mapping. The transfer configuration specifies the directory where the documents are created. The import process does not try to re-create the folder from which the document was exported but instead uses the directory specified in the transfer configuration.

When a record was associated with a document with multiple content elements, the import process recreates all of the exported content elements. Similarly, if multiple documents are associated with a record, all of the associated documents are re-created. These documents are created in the Record Object Store and directory specified in the import mapping.

During the import process, the Records Manager Transfer tool first imports all records folders found in the import directory. If the specified folder already exists, the existing folder is maintained and the information from the imported folder is ignored, which allows the user to successfully perform multiple imports that utilize the same folder.

10.1.3 Lifecycle information

If a record or record folder has reached cutoff and begun the disposition process, the export includes information in the generated XML about the current disposition state of the record or record folder. This information includes the last disposition phase that was executed and when it was executed and the final disposition phase that will be executed. If it can be calculated, it also includes the date when the final phase will occur.

During import, the tool attempts to re-create the disposition status of the record or record folder. The Records Manager Transfer tool expects the fileplan to already exist on the target FPOS. The user can create the fileplan manually, or the user can use the Records Manager File Plan Import Export tool. This tool creates all fileplan elements, including categories and schedules, as well as associated items, such as locations and events.

The tool compares the exported lifecycle information to the schedule on the target container. If the information matches, this information is used to calculate the cutoff date and the current phase, as well as all other disposition information. This information is then set on the imported record or record folder.

If the imported record information does not match the schedule assigned to the target container, the import process does not set the disposition information. In this case, the Records Manager must run the Disposition Sweep program after the import process is complete. Disposition Sweep uses the assigned schedule and record or record folder information to determine the cutoff date and the current phase.

10.1.4 Record to record links

IBM FileNet Records Manager provides a variety of ways to link records. These links define a variety of relationships between records and provide additional key information about a record. They are included in the exported data, which means that they are also available for import.

The schema being used has nodes for two specific types of links: supporting file links and attachment reference links. During the export process, these links are used to output any linked records that are marked as supporting documents or e-mail attachments respectively. During the import, these links are automatically re-created if both the source and target record exist in the target system.

For other links, we have defined an ObjectRef element in the user-defined section of the XML that identifies a link. This element contains the information about the record to which the current record is linked. It is enclosed in another element, which identifies the type of link that is being represented. This design provides enough information about the link for it to be re-created during the import process.

A final special link object is used for holds. A hold is represented by a custom object in IBM FileNet Records Manager. The exported hold link identifies the hold object being referenced instead of another record. With this mechanism, the record hold can be re-created if the hold object exists on the target system. The File Plan Import Export Tool can be used to move hold links from one system to another system.

When a link object is found during the import process, the first step is to determine if the target object exists in the system. If it does, the link is created on the target system. If the linked object does not exist, the link cannot be re-created. In general, when the second record is created later in the import process, it detects that the first object exists and creates the link object.

10.1.5 Audit data

When creating an export mapping, the records manager can set an option to include audit data for the exported objects. This audit trail information tracks all changes made to the object since its creation, including metadata changes. If this option is selected, for each object selected, the generated XML includes a metadata field that holds the audit data for that object. This field is a string field and includes a concatenation of all of the audit events for the object. Be careful about selecting this option, because records that have existed for a long period of time can have very large audit trails, which inflate the size of the exported data. Only select this option if it is critical to include the audit trail information for the exported objects.

Because the audit data is included in a user-defined metadata field, it can be mapped to a metadata field when the import mapping is created. If the audit data field is mapped, the exported audit data is assigned to the specified metadata field on the object, which keeps the applicable audit data associated with the object after it has been imported.

Because the audit log is secured and cannot be modified by external processes, it is not possible to move the imported audit data into the audit trail on the target system. For the imported record, previous audit information is included in the object metadata and not in the audit trail on the new system. Any future events on the imported object are then added to the audit trail on the target system.

10.1.6 Mapping definitions

Mapping definitions are created for both the import and export processes. Creation and maintenance of the mappings are performed through the Records Manager application and are accessed through the Configure page by selecting the Transfer Mappings link, which then displays the mappings maintenance page. This page allows a Records Administrator or Records Manager to create new mappings (import and export) as well as to modify existing mappings. They can also copy and delete existing mappings. The maintenance page displays a list of all currently created mappings.

The mappings are associated with a specific FPOS , because they are dependent on the metadata defined in the fileplan. You can create or modify mappings for an FPOS by setting that Object Store as your default. After you have selected an FPOS, the mapping maintenance page shows the mapping files associated with that FPOS only.

10.1.7 Limitations

The current version of the Records Manager Transfer tool has limitations that prevent a complete transfer of record information. Most of these limitations are due to the limitations of the schema currently being used. Future versions might have more schema extensions that will assist in reducing these limitations.

The first limitation is on the creation of the documents that contain the content elements associated with the records. Currently, the Records Manager Transfer tool does not re-create all information about these documents. For instance, the original document class is lost during the transfer process. In addition, the Records Manager Transfer tool does not re-create the file structure on the target Record Object Store. Instead, all documents and associated content elements are created in a single directory that is specified during import configuration.

Document version information is also lost during the transfer process. For instance, assume that you have a document with three versions and each of these versions is declared as a record. When these documents are imported, each document is created as a separate document instead of creating a single document with three versions. All of the content is intact, but the information that these versions are versions of the same document is lost.

If a record is filed in multiple containers, this information is lost. Only the link to the primary container, which is called the *security parent*, is contained in the exported XML. The information about the other containers is lost. The security folder is set to the first folder in which the record is filed.

Finally, records cannot be transferred from one fileplan to a second fileplan on the same FPOS. Because we reuse the Unique System Identifiers, the import process detects that there are duplicate IDs and stops the import process.

10.2 XML output

During the export, a file is created for each exported record and record folder. The file for a specific record or folder contains all of the exported metadata for that object. For records, this file also contains all of the content elements associated with the record. Information about the content elements is exported that allows the import process to determine whether the content elements are associated with a single document or multiple documents. The import process then re-creates the documents and content associated with the record so that it matches the structure on the source system.

Each file created contains the full Globally Unique Identifier (GUID) associated with the exported object. This GUID is then prepended with either "F_" for a record folder or "R_" for a record. This naming convention allows the user to identify the objects that have been exported.

During the import process, all of the files in the specified directory are imported. All of the record folder files are imported first, because the importation of the records can depend on the folders already existing. Because multiple export jobs might include the same folder, if a folder already exists on the target system, it is not reimported. In addition, to prevent issues when an export set is imported a second time, if a record being imported already exists, it is not reimported. This condition is detected, because we reuse the System Identifiers for imported records.

In addition to the object files, the schema used during the export is generated based on the values in the transfer mapping and is included in the export directory. These schema files can then be used during the import process to map any user-defined or organization-defined fields during the import process.

A Document Type Definition (DTD) file matching the schema is also included in the export directory. A DTD is an alternate to XML schema files for defining the content of XML files. Both the DTD and XML schema files can be used to validate the content of the XML files.

10.3 Export process

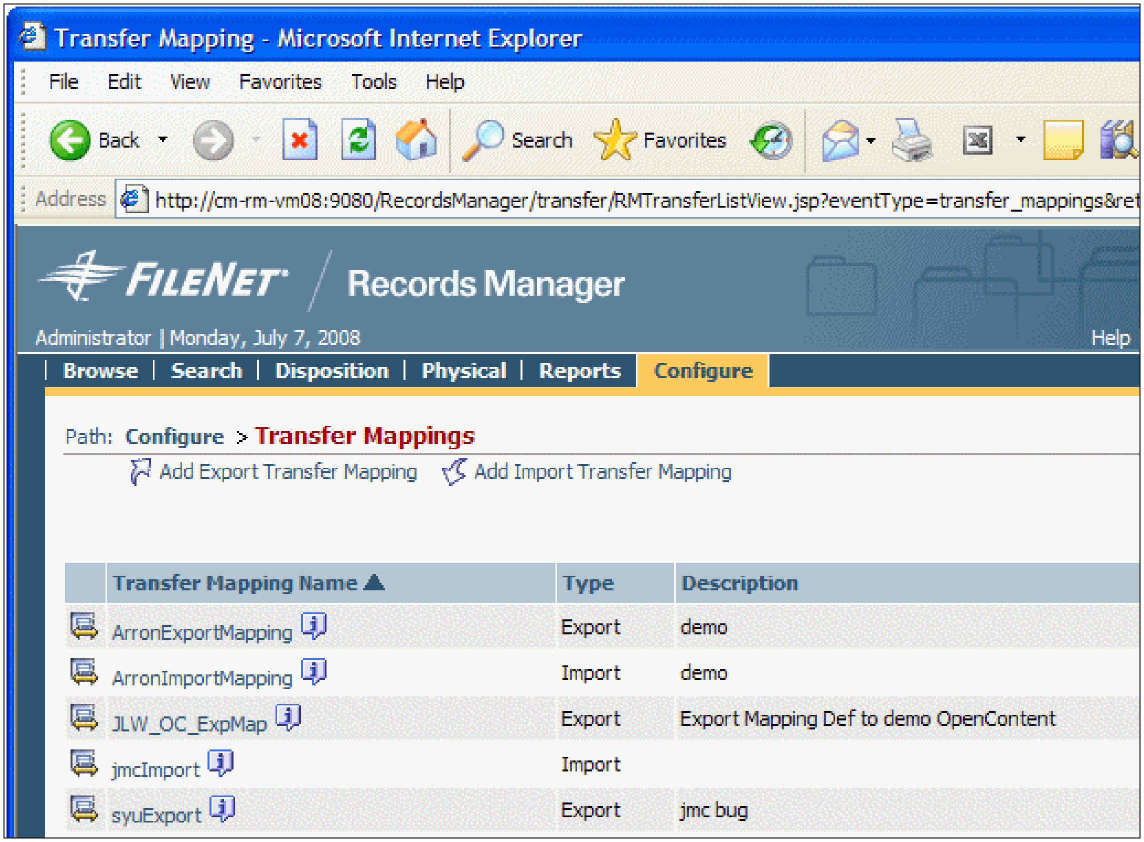
In this section, we describe the process of exporting records. While we describe all of the major steps, we do not provide detailed information or discuss every option available. For more detailed information, refer to the online help for the Records Manager Transfer tool.

10.3.1 Creating an export mapping

An export mapping must be created before an export can occur. The export mapping defines export options and specifies which additional fields are included in the generated XML. An export mapping must be used even if no additional fields are defined.

Export mappings are created in the Export Mappings section of the Records Manager application. You access this functionality by selecting the Transfer Mappings icon from the Configure tab in the Records Manager application. Only Records Managers and System Administrators have access to the functionality.

Figure 10-1 on page 243 shows the initial page for the Transfer Mapping tool. As can be seen, the tool provides a list of all of the currently created mappings and identifies which are import mappings and which are export mappings. In addition, the tool bar at the top of the page provides the capability to create a new import or export mapping.

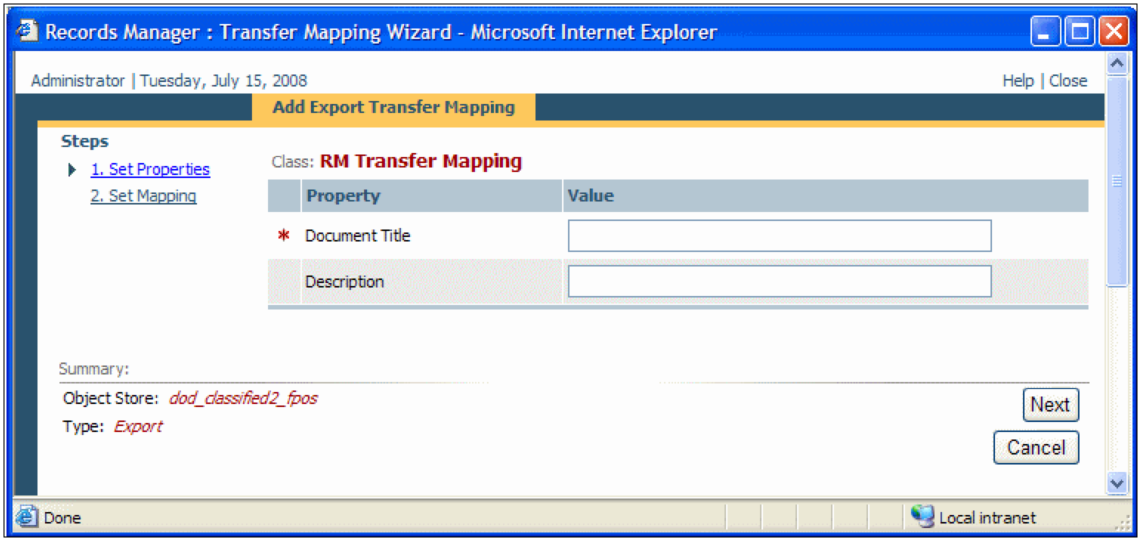

Transfer Mapping - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
http://cm-rm-vm08:9080/RecordsManager/transfer/RMTransferListView.jsp?eventType=transfer_mappings&ret
FILENET
Records Manager
Administrator | Monday, July 7, 2008
Help
Browse | Search | Disposition | Physical | Reports | Configure
Path: Configure > Transfer Mappings
Add Export Transfer Mapping
Add Import Transfer Mapping

Transfer Mapping Name ▲	Type	Description
ArronExportMapping	Export	demo
ArronImportMapping	Import	demo
JLW_OC_ExpMap	Export	Export Mapping Def to demo OpenContent
jmcImport	Import	
syuExport	Export	jmc bug

Figure 10-1 Transfer mapping page shows existing mappings

To create a new export mapping:

1. You select the Create Export Transfer Mapping icon at the top of the page. Figure 10-2 on page 244 shows the first step in the export mapping creation wizard.
2. On this page, you specify a name and optional description for the mapping that you create. Duplicate mapping names are not allowed, so each mapping name must be unique. After you enter this information, you select **Next** to show the second page of the wizard.


Records Manager : Transfer Mapping Wizard - Microsoft Internet Explorer
Administrator | Tuesday, July 15, 2008
Help | Close
Add Export Transfer Mapping
Steps
1. Set Properties
2. Set Mapping
Next
Cancel
Done
Local intranet

Class: RM Transfer Mapping

	Property	Value
*	Document Title	
	Description	

Summary:

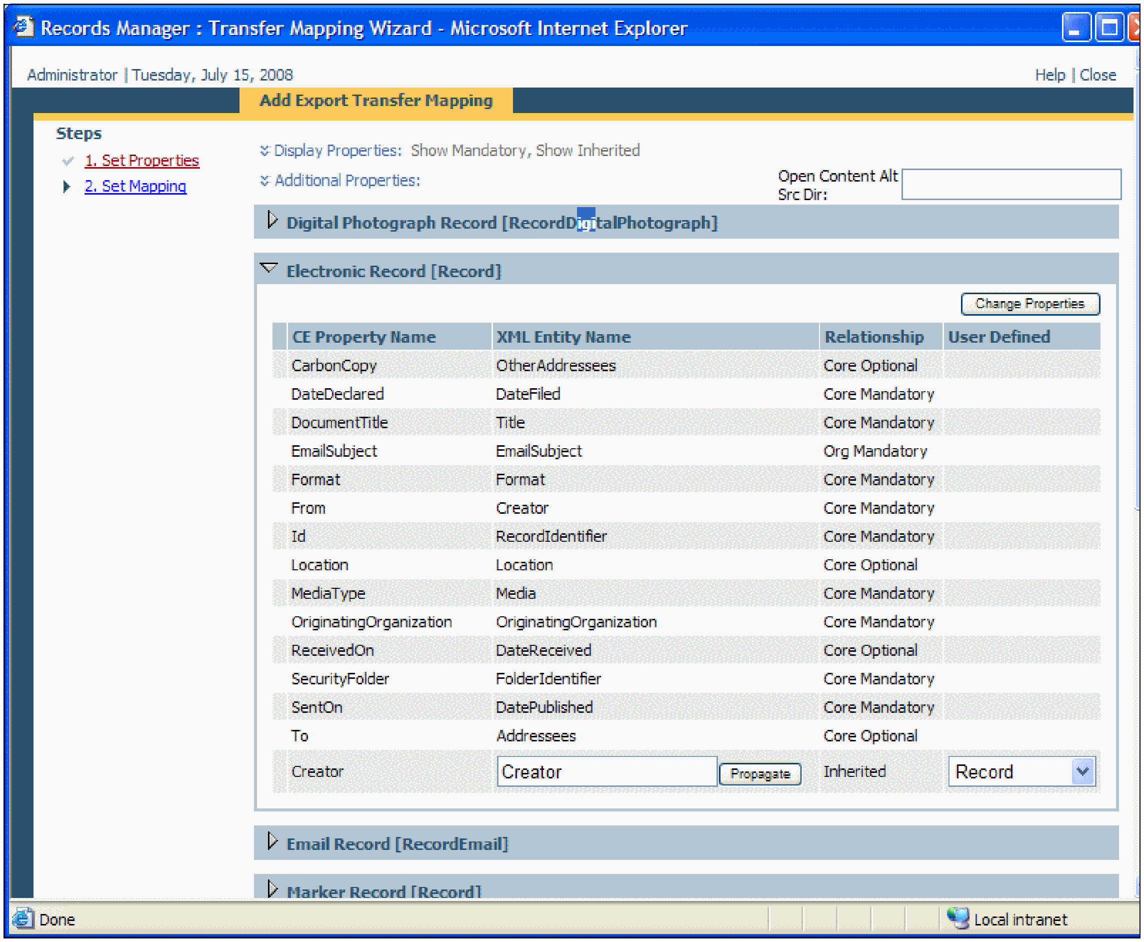
Object Store: *dod_classified2_fpos*

Type: *Export*

Figure 10-2 Initial export mapping page used to identify mapping

3. Figure 10-3 on page 245 shows the mapping page. In this example, the section for Electronic Record has been expanded. The major part of this page shows each of the record classes defined on the system. Expanding one of these classes by selecting the arrow shows the fields for that class. The required fields are automatically mapped and cannot be modified. To add additional fields, you select **Change Properties**. Each added property is assigned a default output name to use in the XML file. For these optional fields, you can modify the name that is used in the XML. In addition, you can specify to which UserDefined area of the schema these optional fields are added. There are user-defined fields in the record, folder, lifecycle, and computer file portions of the schema.

   After you have finished your mappings, you select **Finish** to complete the wizard.



Records Manager : Transfer Mapping Wizard - Microsoft Internet Explorer

Administrator | Tuesday, July 15, 2008

Help | Close

Add Export Transfer Mapping

Steps

1. Set Properties

2. Set Mapping

Display Properties: Show Mandatory, Show Inherited

Additional Properties:

Open Content Alt Src Dir:

Digital Photograph Record [RecordDigitalPhotograph]

Electronic Record [Record]

Change Properties

CE Property Name	XML Entity Name	Relationship	User Defined
CarbonCopy	OtherAddressees	Core Optional	
DateDeclared	DateFiled	Core Mandatory	
DocumentTitle	Title	Core Mandatory	
EmailSubject	EmailSubject	Org Mandatory	
Format	Format	Core Mandatory	
From	Creator	Core Mandatory	
Id	RecordIdentifier	Core Mandatory	
Location	Location	Core Optional	
MediaType	Media	Core Mandatory	
OriginatingOrganization	OriginatingOrganization	Core Mandatory	
ReceivedOn	DateReceived	Core Optional	
SecurityFolder	FolderIdentifier	Core Mandatory	
SentOn	DatePublished	Core Mandatory	
To	Addressees	Core Optional	
Creator	Creator Propagate	Inherited	Record

Email Record [RecordEmail]

Marker Record [Record]

Done

Local intranet

Figure 10-3 Export mapping page is used to define additional mappings

10.3.2 Configuring an export

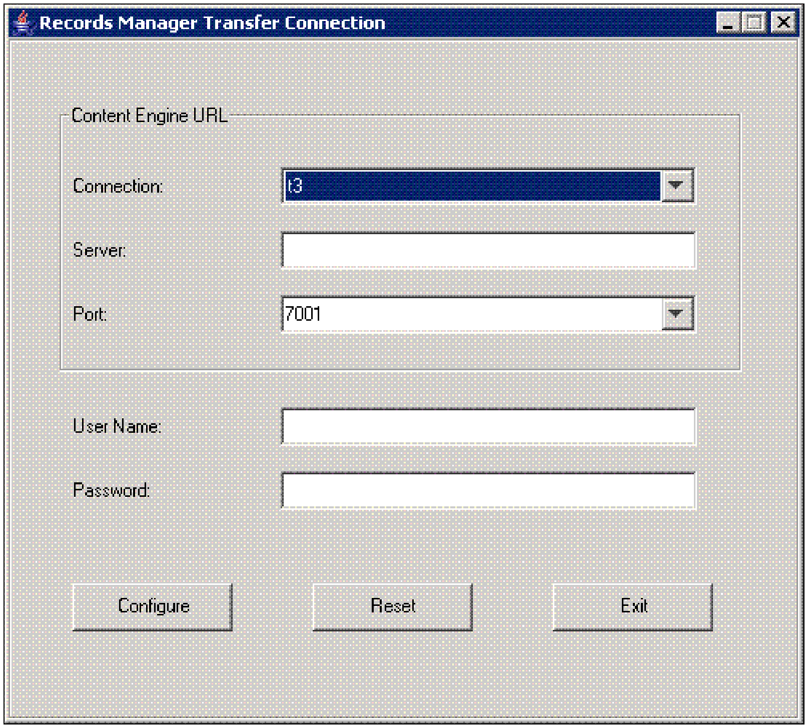
On a fresh installation, the Records Administrator needs to update the RMTransfer.bat file to specify the Java classpath information so that the correct .jars files are located based on the application server being used. Refer to the Records Manager on-line help for information about making these modifications.

Before running export, you must specify the connection and configuration information for the tool:

1. This action is done in two separate steps, because the connection information is typically be used for a series of exports, while the configuration information is modified for each new export.

   To set the connection information for export, open a command prompt on the server and browse to the directory where the Records Manager Transfer tool is installed. You then run the following command (in Windows) that displays the page in Figure 10-4. Other operating systems have a similar command:

   ```
   rmtransfer.bat -configure connection
   ```



Records Manager Transfer Connection

Content Engine URL

Connection: t3

Server:

Port: 7001

User Name:

Password:

Figure 10-4 Configure connection is used to define the server connection for export

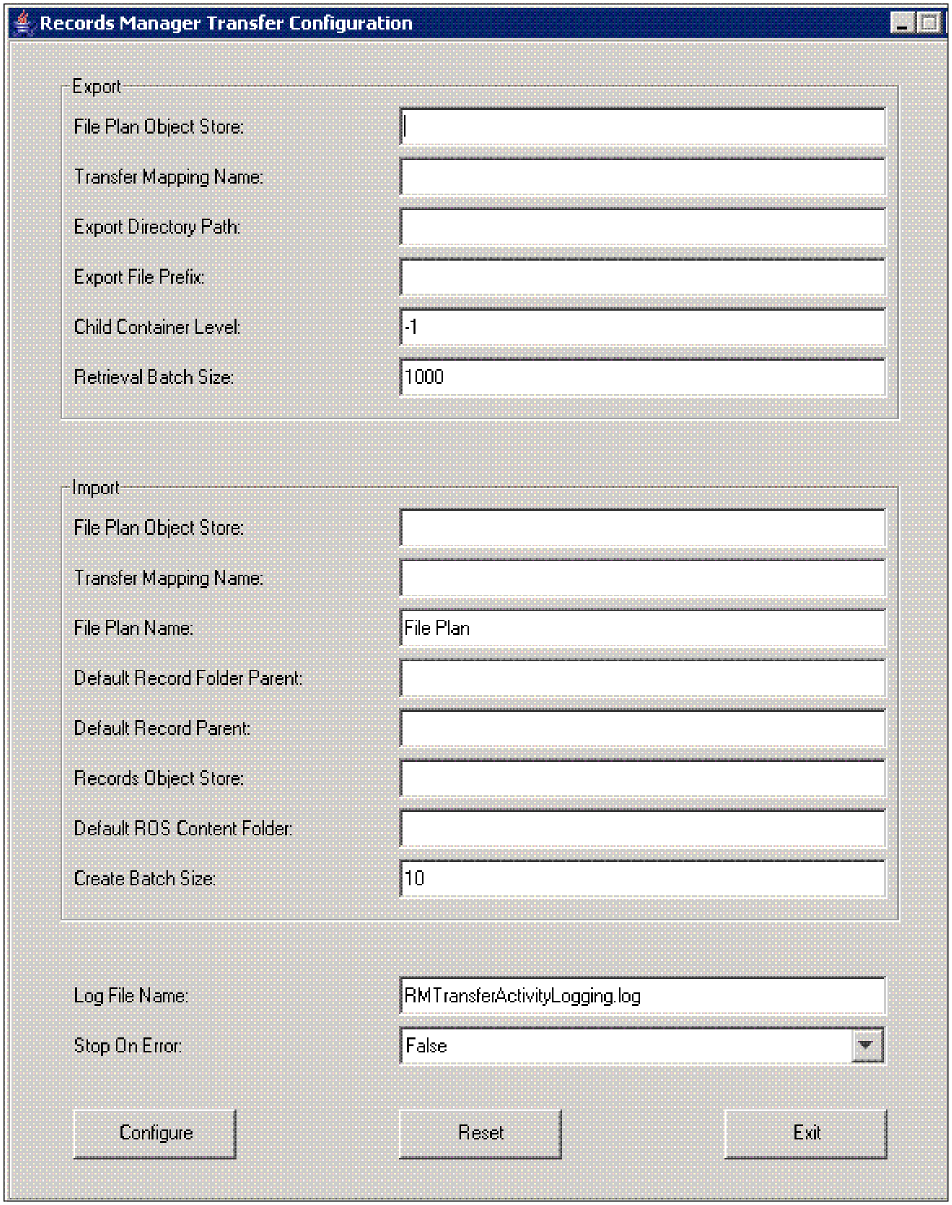
2. Figure 10-4 allows you to enter the information about the Content Engine server where the records reside. The connection field specifies the type of connection to use when communicating with the server. Select the appropriate item from the drop-down list. The Server Name field is where you specify the name of the Content Engine server. The Port field allows you to specify the port to use when communicating with the server. If your system has been configured to use the default port, you can select the appropriate Application Server from the drop-down list. If you are using a non-standard

port, type the port number directly into the field. Finally, enter the Username and Password to use when running the tool.

After all of the values have been specified, click **Configure**, which saves the specified values in a configuration file. For security reasons, the password is encrypted before it is stored in this file.

3. Next, you need to specify the export information by executing the following command (in Windows), which displays the dialog in Figure 10-5 on page 248:

```
rmtransfer.bat -configure configuration
```



Records Manager Transfer Configuration

Export

File Plan Object Store:	
Transfer Mapping Name:	
Export Directory Path:	
Export File Prefix:	
Child Container Level:	-1
Retrieval Batch Size:	1000

Import

File Plan Object Store:	
Transfer Mapping Name:	
File Plan Name:	File Plan
Default Record Folder Parent:	
Default Record Parent:	
Records Object Store:	
Default ROS Content Folder:	
Create Batch Size:	10

Log File Name:	RMTransferActivityLogging.log
Stop On Error:	False

Figure 10-5 Export configuration is used to define export configuration options

4. In Figure 10-5, the top half is used to specify the export information. The FPOS field is used to specify the FPOS where the records to be exported are stored. The Transfer Mapping Name specifies which export mapping to use in the export process. The Export Directory path specifies the directory where

the exported records and records folders are created. This directory must exist before the export is started. The Export File Prefix allows the user to specify a prefix used in each filename that is created. The child container level allows you to limit the number of levels in the fileplan that are exported. Specifying -1 aligns the whole tree beneath the specified export item. Finally, the Retrieval Batch Size is used to determine how many records are retrieved at a time. This parameter is used for performance tuning.

10.3.3 Executing an export

The final step in the export process is to perform the actual export. Perform this step after the Mapping is created and both the Connection information and the Configuration information have been specified.

Like the configure steps, executing the Records Manager Transfer tool is done on the server where it is installed. You again browse to the directory and then execute the export command. For export, there are a variety of ways to specify the items to export. You can specify specific records, folders, or categories, or you can specify an input file that contains the items to be exported. When specifying entities, you can use either their full path names or you can use their system unique IDs. The following command gives an example of executing the export process for a specified category:

```
rmtransfer.bat -export -category {FFDDF596-3D15-4F72-BF68-804002E78CC6}
```

The Records Manager Transfer tool outputs information while the tool runs, and then it prints a summary of the number of records and records folders that were exported. It also identifies if any errors occurred during the export process. After completion of the export, the exported records and records folders are located in the directory specified in the configuration dialog.

10.4 Import process

As expected, the import process is used to import records and records folders that were previously exported. Usually, these entities are from a different FPOS. Before importing the records, insure that the appropriate fileplan information exists in the target repository, including the categories and subcategories needed for the records and records folders. In addition, you need to make sure that the Record Classes that are used by the imported records exist on the target system.

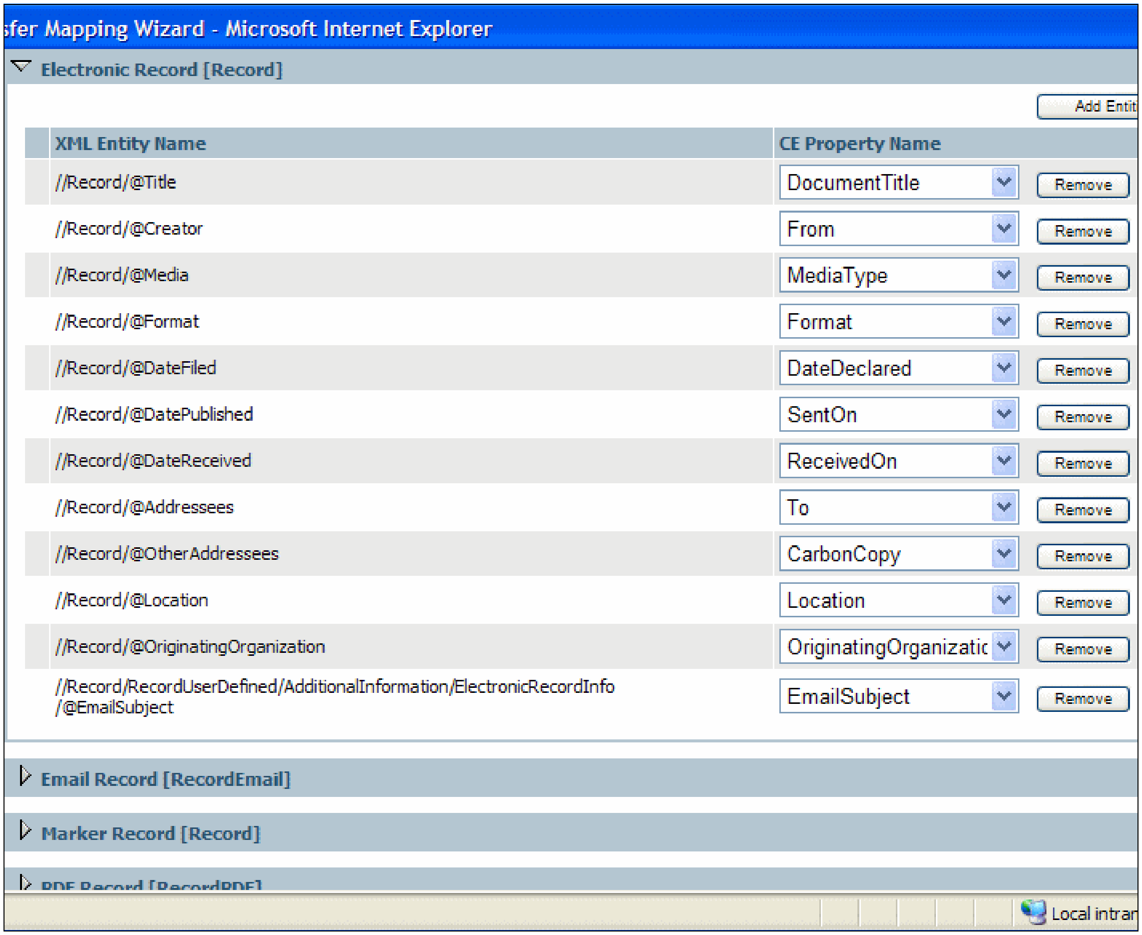
10.4.1 Creating an import mapping

Like export, the first step in the process is to create an import mapping. An import mapping is separate from an export mapping and contains separate information. The import mapping is used to specify to what fields the imported data is mapped by reading the schema associated with the imported data and allowing the user to map any schema fields to existing fields on the FPOS.

Just like export, the import mapping is created in the Transfer Mappings section of the Configuration tab.

The steps are:

1. After you have selected Create Import Transfer Mapping, a wizard begins. The initial dialog is the same as the export page that is shown in Figure 10-2 on page 244.
2. In the second step of the wizard, you must upload the schema files to the server. The schema files describe the format of the XML files that are to be imported. They are based on the Joint Interoperability Test Center (JITC) standard and specify which fields are included in the XML that is to be imported. After you upload the schema files, you select **Next** to go to the final page of the wizard.
3. Figure 10-6 on page 251 shows the final page of the import mappings wizard. In this page, all of the required fields in the schema are automatically mapped. You have the option of adding additional fields by selecting **Add Properties** for a specific document class, which brings up a tree view that represents all of the fields defined in the import XML. Any of these fields can be added to the mapping page by selecting the object. On the mapping page, you can then select to which Content Engine (CE) field this new value is mapped, which allows you to map all of the data contained in the import XML for inclusion when the imported records are created. After you have completed mapping the fields, you complete the wizard by clicking **Finish**.


sfer Mapping Wizard - Microsoft Internet Explorer
Electronic Record [Record]
Add Entit
Remove
Email Record [RecordEmail]
Marker Record [Record]
PDF Record [RecordPDF]
Local intrar

XML Entity Name	CE Property Name
//Record/@Title	DocumentTitle
//Record/@Creator	From
//Record/@Media	MediaType
//Record/@Format	Format
//Record/@DateFiled	DateDeclared
//Record/@DatePublished	SentOn
//Record/@DateReceived	ReceivedOn
//Record/@Addressees	To
//Record/@OtherAddressees	CarbonCopy
//Record/@Location	Location
//Record/@OriginatingOrganization	OriginatingOrganizatic
//Record/RecordUserDefined/AdditionalInformation/ElectronicRecordInfo /@EmailSubject	EmailSubject

Figure 10-6 Import mapping page used to map schema items to CE fields

10.4.2 Configuring an import

Configuring an import follows the same steps as configuring an export. The connection option uses the exact same options as for export. On the configuration page, you need to enter the import information at the bottom of the dialog. This dialog is shown in Figure 10-5 on page 248.

As with export, you start by specifying the FPOS where the records and records folders are to be created. You then specify the import transfer mapping that specifies the mapping from the import files to the Content Engine fields. You then specify the name of the fileplan to be used. The next two fields specify where records and records folders are to be created if their parent does not exist in the target FPOS. Next, you specify the Records-enabled content Object Store (ROS) and the ROS folder where the content needs to be created. Finally, you specify the create batch size. This parameter is typically left at the default value unless tuning indicates that a different value provides higher throughput.

10.4.3 Executing an import

To perform the import, you must run the import tool, which is typically located on the CE or RM server. Traverse to the file location where RM is installed and then go to the RMTransfer directory.

For import, the main option is the directory where the import files are located. All of the files in this directory are imported. When running import, ensure that the import mapping file used matches the FPOS where the records are created. A typical import command for Windows is:

```
rmtransfer.bat -import -dir “c:\myimportdir”
```

Part 2

Implementation with case study

In this part, using a case study, we provide step-by-step instructions how to implement a sample records management solution. We intend to provide concrete examples of how to perform the tasks, including file plan creation, records ingestion and declaration, record disposal, record hold, and sample programs using IBM FileNet Records Manager application programming interfaces (APIs).

11

File plan creation case study

In this chapter, we describe the steps required to records-enable an IBM FileNet P8 system.

We discuss the following topics in this chapter:

- Introducing the case study file plan
- Preparing object stores
- Building case study-specific data model objects
- Configuring Workplace site preferences
- Creating a file plan:
  - Build the file plan categories
  - Creating an electronic record folder

> **Note:** This chapter does not provide the details of installing IBM FileNet Records Manager or the details of performing the standard Content Engine functions, such as creating object stores and defining document classes. For these instructions, refer to ecm_help.

11.1 Introducing the case study file plan

Our case study is based on the fictitious Fictional Insurance Company X that we introduced in 3.3.2, “Example file plan” on page 65. The company’s file plan (called the XYZ File Plan) is shown in Figure 3-3 on page 66. The partial file plan showing categories that are related to the case study for this book is illustrated in Figure 11-1. This file plan is designed to showcase:

- A variety of ingestion and record declaration options that IBM FileNet Records Manager offers
- The impact of various disposition aggregation levels on disposition and hold processing

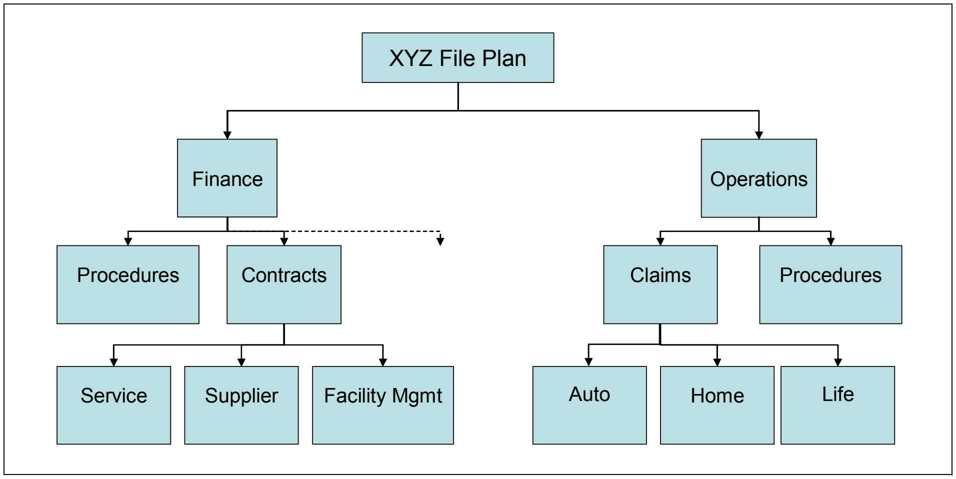


Figure 11-1 Partial file plan showing categories related to the case study for this book

The case study focuses on three areas of the file plan:

- Contracts within Finance:
  - Declaration of records using entry templates
  - Record level aggregation
- Claims within Operations:
  - Declaration of records through workflow
  - Folder level aggregation
- Procedures across all functional areas:
  - Declaration of records using Content Engine (CE) events

As best practice dictates, File Plan Object Stores (FPOS) and Records-enabled content Object Store (ROS) must not be collocated. For this case study, we create two separate object stores:

- RedBook_Records (FPOS)
- RedBook_Documents (ROS)

Records enabling an IBM FileNet P8 system require a thorough understanding of IBM Filenet Enterprise Manager, as well as the IBM FileNet Records Manager application.

This chapter does not provide the basic installation instructions for the IBM FileNet P8 products. We assume that you have a working version of IBM FileNet P8 system (either an IBM FileNet Content Manager or IBM FileNet Business Process Manager system) with IBM FileNet Records Manager. The steps that we provide in this chapter are required to be performed each time that you need to create a new set of object stores for use with IBM FileNet Records Manager.

11.2 Preparing object stores

Before a file plan can be created in IBM FileNet Records Manager, the object stores that will be used either as FPOSs or a ROS need to have their data models updated with the required IBM FileNet Records Manager data models.

IBM FileNet Records Manager supports four data model options:

- Base
- Department of Defense (DoD)
- DoD Classified
- PRO

For the case study, we use the most common data model, the Base data model.

There is a a set of XML files and Visual Basic® (VB) scripts that need to be imported into the appropriate object stores to update your data model. These XML files and scripts update the existing object store data model with property templates, document classes, and custom classes, among other items, that are required by the IBM FileNet Records Manager application and application programming interfaces (APIs).

To prepare the object stores for the file plan, perform the following steps:

1. Importing data models into the object stores.
2. Importing and setting up security templates.
3. Transferring IBM FileNet Records Manager workflow definitions.

11.2.1 Importing data models into the object stores

To import IBM FileNet Records Manager data models into the object stores, perform the following tasks:

- Importing the File Plan Object Store data model (to the FPOS).
- Importing the file plan reports (to the FPOS).
- Importing the Records-enable content Object Store data model (to the ROS).

The XML files and scripts that we use to import data models are provided as part of the installation for IBM FileNet Records Manager; however, they are not normally copied over as part of the installation. They normally exist in a similar directory structure as shown in Figure 11-2.

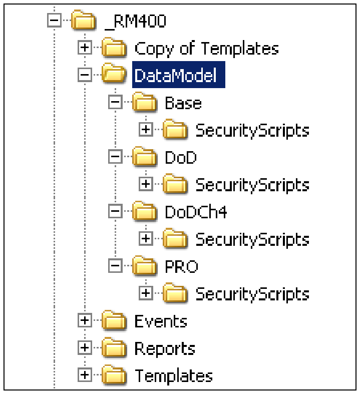
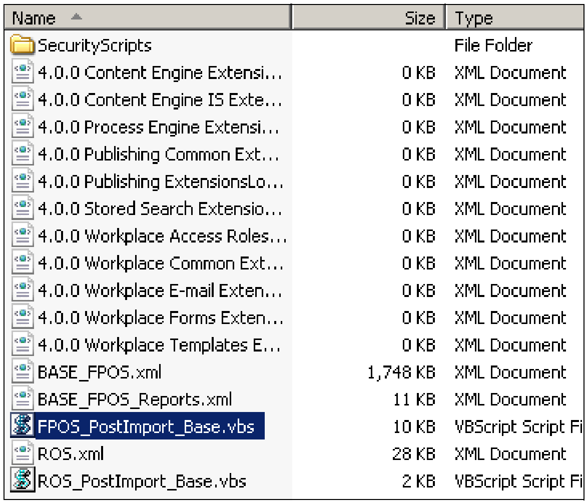


Figure 11-2 IBM FileNet Records Manager configuration directory

To import the data models into the object stores, perform the importing tasks on a machine that has IBM Filenet Enterprise Manager installed and that is configured to access your IBM FileNet P8 domain.

Importing the File Plan Object Store data model

The XML file, BASE_FPOS.xml, is used to import the File Plan Object Store data model. The file is located in the ..\DataModel\Base directory. Refer to Figure 11-3 on page 259.



Name	Size	Type
SecurityScripts		File Folder
4.0.0 Content Engine Extensi...	0 KB	XML Document
4.0.0 Content Engine IS Exte...	0 KB	XML Document
4.0.0 Process Engine Extensi...	0 KB	XML Document
4.0.0 Publishing Common Ext...	0 KB	XML Document
4.0.0 Publishing ExtensionsLo...	0 KB	XML Document
4.0.0 Stored Search Extensio...	0 KB	XML Document
4.0.0 Workplace Access Roles...	0 KB	XML Document
4.0.0 Workplace Common Ext...	0 KB	XML Document
4.0.0 Workplace E-mail Exten...	0 KB	XML Document
4.0.0 Workplace Forms Exten...	0 KB	XML Document
4.0.0 Workplace Templates E...	0 KB	XML Document
BASE_FPOS.xml	1,748 KB	XML Document
BASE_FPOS_Reports.xml	11 KB	XML Document
FPOS_PostImport_Base.vbs	10 KB	VBScript Script Fi
ROS.xml	28 KB	XML Document
ROS_PostImport_Base.vbs	2 KB	VBScript Script Fi

Figure 11-3 Base data model files

Use the following setup when importing the File Plan Object Store data model:

- Object Store: RedBook_Records (for our case study)
- Import Options tab:
  - Import Manifest File: ..\DataModel\Base\BASE_FPOS.xml
  - Retry Failed Imports: Selected
- Scripts tab:
  - Run Type: Post-Import
  - XML Import: FROS_PostImport_Base.vbs

To import the File Plan Object Store data model, follow these steps:

1. Launch IBM Filenet Enterprise Manager and log on as the IBM FileNet P8 Domain Administrator.
2. Expand the list of object stores.
3. Right-click your File Plan Object Store and select **All Tasks** → **Import All** from the context menu.

   For our case study, we use RedBook_Records as the File Plan Object Store. Refer to Figure 11-4 on page 260.

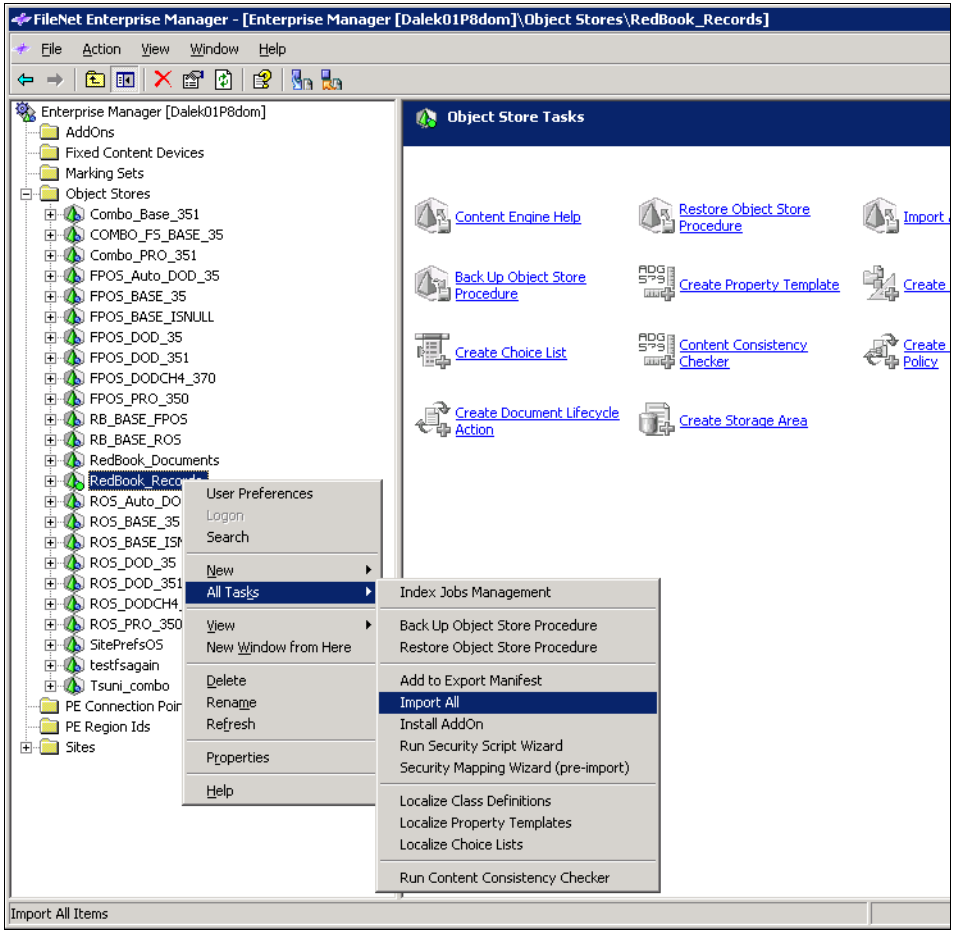


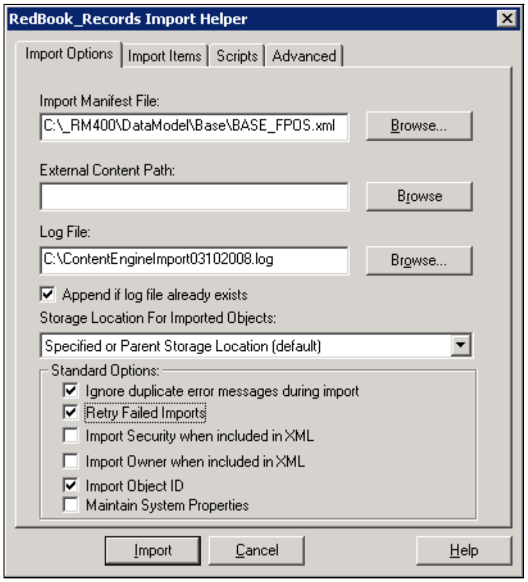
Figure 11-4 Import All menu option

4. Set the Import Options (refer to Figure 11-5 on page 261):

   a. Select **BASE_FPOS.xml** for the Import Manifest File entry box.

      In our case study, the file is located in ..\DataModel\Base directory. You can also use **Browse** to navigate to where the XML file is located and select it.

   b. Select **Retry Failed Import** under Standard Options, which ensures that any failures during import are retried.



Import Manifest File: C:_RM400\DataModel\Base\BASE_FPOS.xml

External Content Path:

Log File: C:\ContentEngineImport03102008.log

☑ Append if log file already exists

Storage Location For Imported Objects: Specified or Parent Storage Location (default)

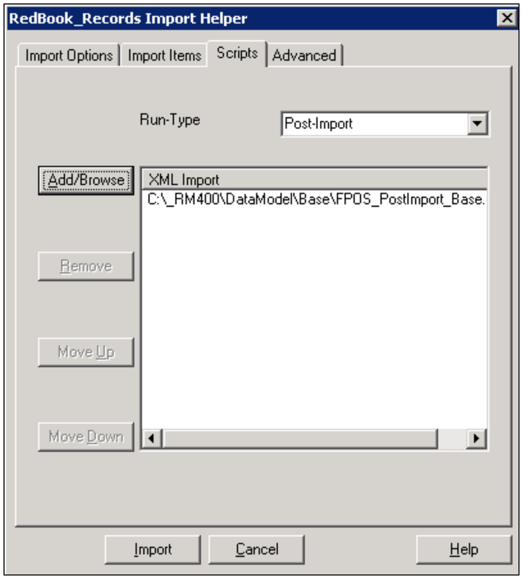
Standard Options:

☑ Ignore duplicate error messages during import
☑ Retry Failed Imports
☐ Import Security when included in XML
☐ Import Owner when included in XML
☑ Import Object ID
☐ Maintain System Properties

Figure 11-5 Import Options tab: Import BASE_FPOS.xml file

5. Set the Post-Import script (refer to Figure 11-6 on page 262):
   a. Select the **Scripts** tab.
   b. Select **Post-Import** from the Run-Type drop-down list.
   c. Click **Add/Browse** and select the **FPOS_PostImport_Base.vbs** file.

      This file is typically located in the same directory where the BASE_FPOS.xml file resides (for example, ..\DataModel\Base).



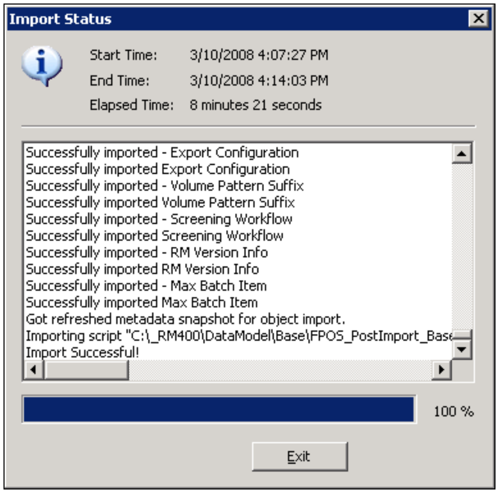
Run-Type: Post-Import

XML Import
C:_RM400\DataModel\Base\FPOS_PostImport_Base.

Figure 11-6 Scripts tab: Adding script to run after the import

6. Click **Import.**

   It is likely to take few minutes to perform the import. On the successful import of the BASE_FPOS data model, a window similar to Figure 11-7 appears.



Start Time: 3/10/2008 4:07:27 PM
End Time: 3/10/2008 4:14:03 PM
Elapsed Time: 8 minutes 21 seconds

```
Successfully imported - Export Configuration
Successfully imported Export Configuration
Successfully imported - Volume Pattern Suffix
Successfully imported Volume Pattern Suffix
Successfully imported - Screening Workflow
Successfully imported Screening Workflow
Successfully imported - RM Version Info
Successfully imported RM Version Info
Successfully imported - Max Batch Item
Successfully imported Max Batch Item
Got refreshed metadata snapshot for object import.
Importing script "C:\_RM400\DataModel\Base\FPOS_PostImport_Base
Import Successful!
```

100 %

Figure 11-7 Successful completion of the base data model import

7. Click **Exit**. Leave IBM Filenet Enterprise Manager open.

Importing the file plan reports

To import the file plan reports into your File Plan Object Store, repeat the previous steps with the following setup:

- Object Store: RedBook_Records (for our case study)
- Import Options tab:
  - Import Manifest File: ..\DataModel\Base\BASE_FPOS_Reports.xml
  - Retry Failed Imports: Selected
- Scripts tab:
  - Run-Type: Blank
  - XML Import: Blank

This BASE_FPOS_Reports.xml file typically is in the same place as the BASE_FPOS.xml file (for example, ..\DataModel\Base directory).

To remove any existing entry in XML Import field in the Scripts tab, select the existing entry and click **Remove**.

Importing the Records-enable content Object Store data model

In order for documents to be declared as records, the documents must reside in a Records-enabled content Object Store. To records-enable an object store to be a ROS, import the data model that is the same type (Base model) that you imported into the FPOS.

To import the data model into the ROS, repeat the steps outlined in “Importing the File Plan Object Store data model” on page 258 with the following setup:

- Object Store: RedBook_Documents (for our case study)
- Import Options tab:
  - Import Manifest File: ..\DataModel\Base\ROS.xml
  - Retry Failed Imports: Selected
- Scripts tab:
  - Run-Type: Post-Import
  - XML Import: ROS_PostImport_Base.vbs

11.2.2 Importing and setting up security templates

IBM FileNet Records Manager provides an enhanced security model that uses roles as the mechanism to access and perform functions within a system.

The default security roles are:

- Records Administrator
- Records Manager
- Records Privileged User
- Records User

Importing user groups and assigning user groups to these roles are performed within IBM FileNet Enterprise Manager:

1. Launch IBM Filenet Enterprise Manager and log on as the IBM FileNet P8 Domain Administrator.
2. Expand the list of object stores.
3. Right-click your File Plan Object Store and select **All Tasks** → **Run Security Script Wizard** from the context menu.

   For our case study, our File Plan Object Store is RedBook_Records. Refer to Figure 11-8 on page 265.

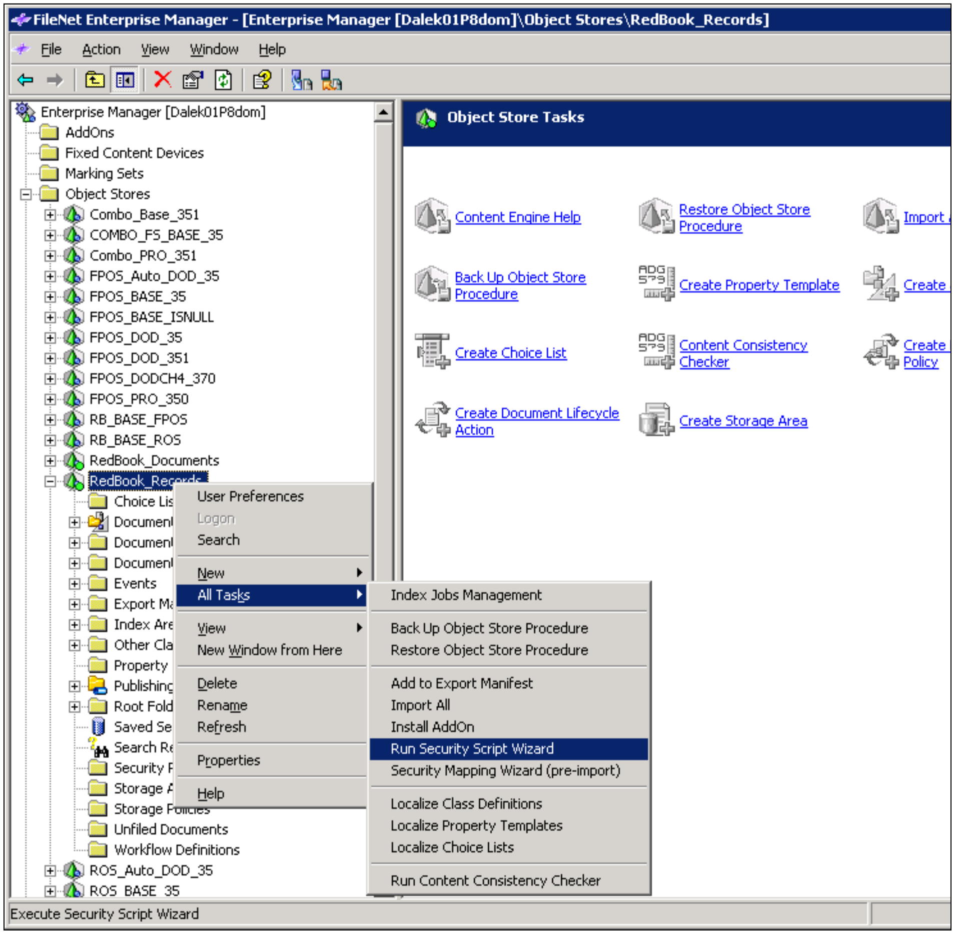
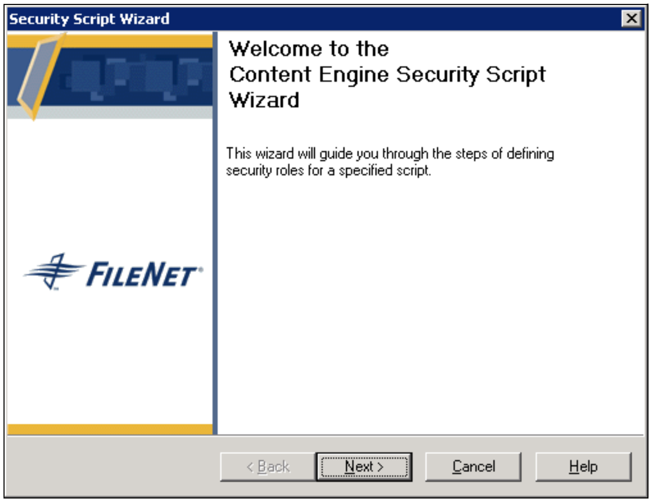


Figure 11-8 Accessing Security Script Wizard task

4. Select **Next** on the Welcome page. Refer to Figure 11-9 on page 266.



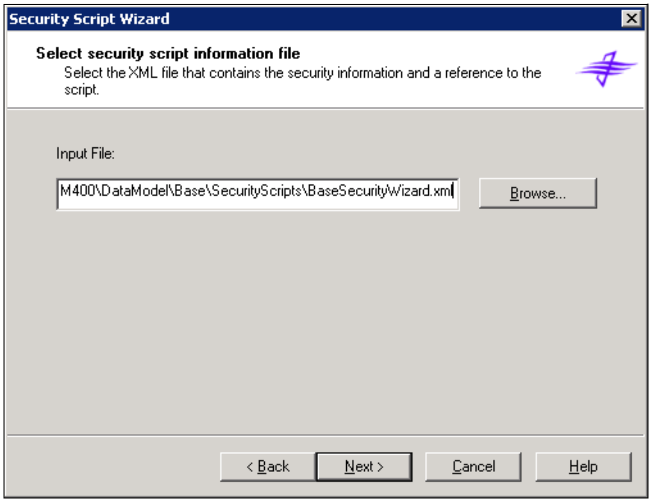
Welcome to the Content Engine Security Script Wizard

This wizard will guide you through the steps of defining security roles for a specified script.

Figure 11-9 Security Wizard Welcome window

5. Select the BaseSecurityWizard.xml file as the Input File and click **Next**. Refer to Figure 11-10.

   This file is normally in the ..\SecurityScripts subdirectory of the data model directory that contains the data model XML files that you imported in the earlier steps.



Select security script information file

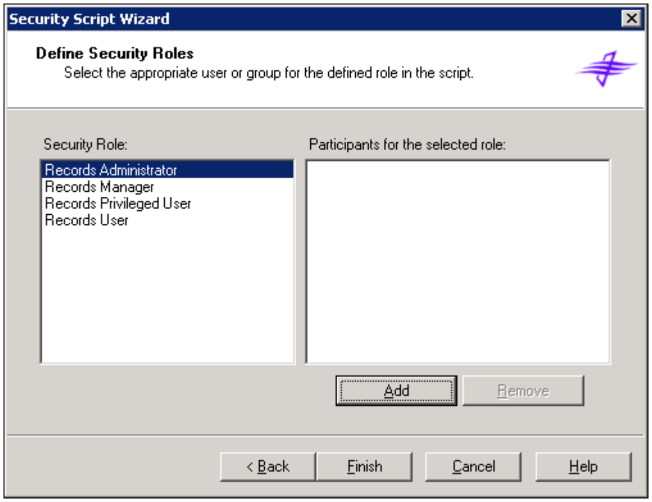
Select the XML file that contains the security information and a reference to the script.

Input File: M400\DataModel\Base\SecurityScripts\BaseSecurityWizard.xml

Figure 11-10 Enter Security Wizard XML file

6. Assign participants to security roles.

   Figure 11-11 shows the Define Security Roles dialog box. You need to assign each role an appropriate Lightweight Directory Access Protocol (LDAP) group (or user). The best practice is to always assign groups rather than individual users to the roles, which provides greater flexibility. You will not then need to change the settings here if a user leaves or enters a group.


Security Script Wizard

Define Security Roles

Select the appropriate user or group for the defined role in the script.

Security Role:

- Records Administrator
- Records Manager
- Records Privileged User
- Records User

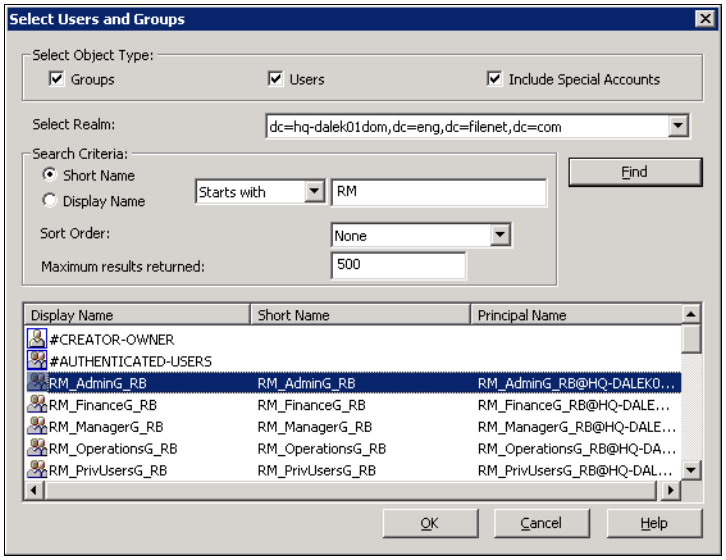
Participants for the selected role:

Figure 11-11 Define Security Roles

   To assign groups to roles:

   a. Select a security role (for example, Records Administrator), and click **Add.**

      The Select Users and Groups dialog box opens. Refer to Figure 11-12 on page 268.



Select Users and Groups

Select Object Type: ☑ Groups ☑ Users ☑ Include Special Accounts

Select Realm: dc=hq-dalek01dom,dc=eng,dc=filenet,dc=com

Search Criteria: ● Short Name ○ Display Name — Starts with — RM

Sort Order: None

Maximum results returned: 500

Display Name	Short Name	Principal Name
#CREATOR-OWNER		
#AUTHENTICATED-USERS		
RM_AdminG_RB	RM_AdminG_RB	RM_AdminG_RB@HQ-DALEK0...
RM_FinanceG_RB	RM_FinanceG_RB	RM_FinanceG_RB@HQ-DALE...
RM_ManagerG_RB	RM_ManagerG_RB	RM_ManagerG_RB@HQ-DALE...
RM_OperationsG_RB	RM_OperationsG_RB	RM_OperationsG_RB@HQ-DA...
RM_PrivUsersG_RB	RM_PrivUsersG_RB	RM_PrivUsersG_RB@HQ-DAL...

Figure 11-12 Select Users and Groups dialog

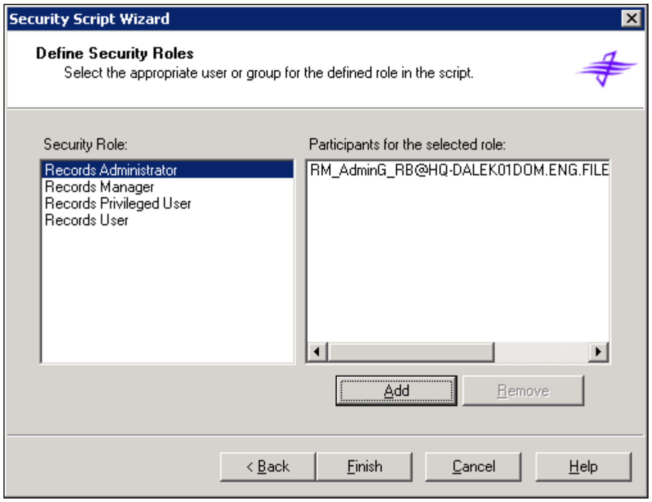
b. Search for the groups to which you want to associate the appropriate role. For our case study, we created the following groups for the appropriate roles:

- RM_AdminG_RB (group) → Records Administrator (role)
- RM_ManagersG_RB (group) → Records Manager (role)
- RM_PrivUsersG_RB (group) → Records Privileged User (role)
- RM_UsersG_RB (group) → Records User (role)

Select the appropriate groups for the current role. If required, multiple groups can be selected by pressing Ctrl when selecting groups.

c. Click **OK** when finished.

The Define Security Roles dialog box shows the group that you assigned to the security role. Refer to Figure 11-13 on page 269 for our case study example.


Security Script Wizard

Define Security Roles

Select the appropriate user or group for the defined role in the script.

Security Role:

- Records Administrator
- Records Manager
- Records Privileged User
- Records User

Participants for the selected role:

RM_AdminG_RB@HQ-DALEK01DOM.ENG.FILE

Figure 11-13 Define Security Roles dialog

d. Repeat the previous steps for each security role.

e. When finished, click **Finish**.

7. Click **OK** on the information window.

   The Wizard informs you that it is applying the security to your file plan. Refer to Figure 11-14.

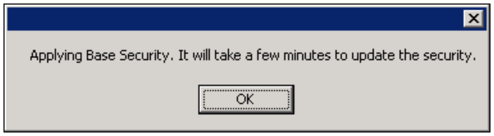
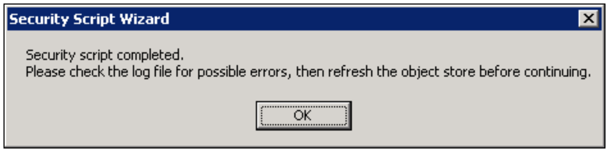

Applying Base Security. It will take a few minutes to update the security.

Figure 11-14 Applying Base security

8. Click **OK** when the process completes. Refer to Figure 11-15.


Security Script Wizard

Security script completed.
Please check the log file for possible errors, then refresh the object store before continuing.

Figure 11-15 Security wizard completed dialog

If you get an error message, the script did not execute properly. Correct any problems that are encountered and rerun the script before proceeding. To get more information about the script execution, refer to the Security Script Wizard log file (BaseSecurity.log) in the ..\DafaModel\Base\SecurityScripts directory. If the wizard is successful, the entries shown in Example 11-1 are at the end of the log file.

Example 11-1 Messages output when script completes successfully

```
3/08/2008 4:40:37 PM : [INFO] The Security Script Wizard has completed
successfully!
3/08/2008 4:40:37 PM : [INFO] Security Script Wizard Exit!
```

11.2.3 Transferring IBM FileNet Records Manager workflow definitions

The disposition workflows that are provided with IBM FileNet Records Manager are installed in the FPOS as part of the data model import step. Before disposition schedules can be created, these workflows need to be transferred to a Business Process Manager (BPM) process region to create an executable version of the process manually by using IBM Filenet Enterprise Manager or by using a batch process.

In this section, we detail the steps to transfer the workflows using the batch process. The process is well documented in the IBM FileNet Records Manager Installation Guide; however, for the completeness of this case study discussion, we provide the steps here.

To perform the following steps, you need to be working on the Application Engine machine on which IBM FileNet Records Manager is installed.

The Workflow Transfer utility and XML configuration file (RMWorkflowTransferConfig.xml) that are used in this procedure are available in the <*RM_install_path*>/FileNet/RM/Workflow/configureRMworkflow folder. For our case study, the directory is C:\Program Files\FileNet\RM\Workflow\configureRMworkflow.

To transfer IBM FileNet Records Manager workflow definitions to a BPM process region:

1. Update the RMWorkflowTransferConfig.xml with the environment-specific setup:
   a. Remove the read-only attribute from the file.
   b. Open RMWorkflowTransferConfig.xml with an XML or text editor.
   c. Update the XML node values as listed in Table 11-1 on page 271.

Table 11-1 XML nodes that need to be updated in RMWorkflowTransferConfig.xml

XML node	Description	Case study value
ce_host_name	The Content Engine server name or IP address	hq-dalek01
ce_wsi_port_number	The Web Services Interoperability (WSI) Data Service port number	8080
connection_point_name	The name of the connection point used in the Application Engine	vwrouter_CP1
object_store_name	The name of the FPOS	RedBook_Records

Example 11-2 shows the updated RMWorkflowTransferConfig.xml that we use in the case study.

Example 11-2 RMWorkflowTransferConfig.xml

```
<workflowtransfer_config>
   <!-- CE Server Name-->
   <ce_host_name>hq-dalek01</ce_host_name>

   <!-- CE WSI Port Number-->
   <ce_wsi_port_number>8080</ce_wsi_port_number>

   <!-- Connection Point Name, which is configured for Workplace/RM-->
   <connection_point_name>vwrouter_CP1</connection_point_name>

   <!-- Object Store Name That Contains The Workflow Definition
Documents-->
   <object_store_name>RedBook_Records</object_store_name>

</workflowtransfer_config>
```

   d. Save and close the RMWorkflowTransferConfig.xml file.

2. Run the Workflow Transfer utility:

   a. Open a command window (DOS window or xTerm).

   b. Change the directory to <*RM_install_path*>/FileNet/RM/Workflow/configureRMworkflow

   c. Run the following command:

      Windows: **WorkflowTransfer.bat *<userid> <password>***

UNIX®: **`./WorkflowTransfer.sh`** ***`<user ID> <password>`***

Where <*userid*> and <*password*> are the user name and password of a user with the object store administrator role of the FPOS object store.

For our case study, we execute:

```
WorkflowTransfer.bat administrator filenet
```

The status of the transfer process is shown in the command window. It is also written in the WorkflowTransfer.log file located at the <*RM_install_path*>/FileNet/RM/Workflow/configureRMworkflow directory.

11.3 Building case study-specific data model objects

For case study-specific data, we need to build data model objects, including choice lists, property templates, document classes, and custom objects in both the ROS (RedBook_Documents) and the FPOS (RedBook_Records).

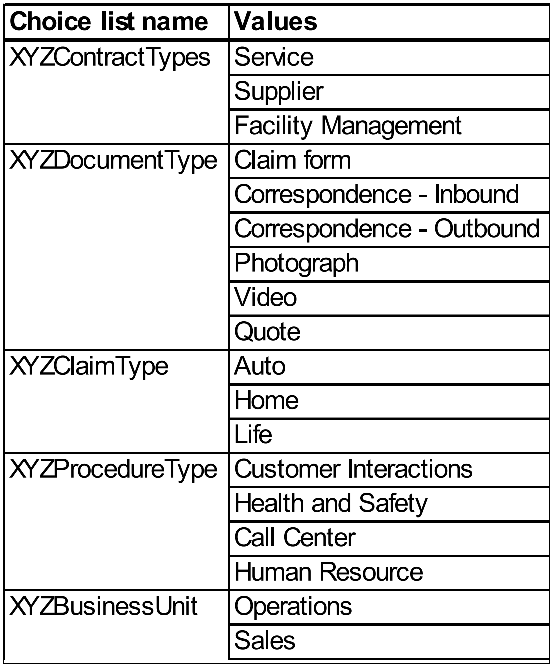
Because these actions are general IBM FileNet Enterprise Manager operations, we do not provide step-by-step instructions for how to create them. In this section, we list the data model objects and their setup information.

11.3.1 RedBook_Documents object store (ROS) setup

The RedBook_Documents object store (ROS) contains content that can be declared as records. We need to add choice lists, property templates, and document classes in the ROS for our case study.

Choice lists

Figure 11-16 on page 273 shows the choice lists that we create for the RedBook_Documents object store.

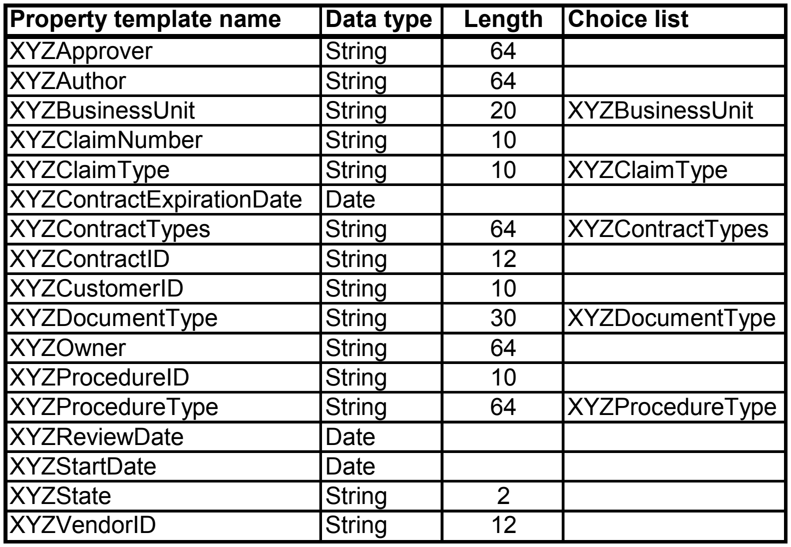


Choice list name	Values
XYZContractTypes	Service
	Supplier
	Facility Management
XYZDocumentType	Claim form
	Correspondence - Inbound
	Correspondence - Outbound
	Photograph
	Video
	Quote
XYZClaimType	Auto
	Home
	Life
XYZProcedureType	Customer Interactions
	Health and Safety
	Call Center
	Human Resource
XYZBusinessUnit	Operations
	Sales

Figure 11-16 RedBook_Documents choice lists

Property templates

Figure 11-17 on page 274 shows the property templates that we create for the RedBook_Documents object store.



Property template name	Data type	Length	Choice list
XYZApprover	String	64	
XYZAuthor	String	64	
XYZBusinessUnit	String	20	XYZBusinessUnit
XYZClaimNumber	String	10	
XYZClaimType	String	10	XYZClaimType
XYZContractExpirationDate	Date		
XYZContractTypes	String	64	XYZContractTypes
XYZContractID	String	12	
XYZCustomerID	String	10	
XYZDocumentType	String	30	XYZDocumentType
XYZOwner	String	64	
XYZProcedureID	String	10	
XYZProcedureType	String	64	XYZProcedureType
XYZReviewDate	Date		
XYZStartDate	Date		
XYZState	String	2	
XYZVendorID	String	12	

Figure 11-17 RedBook_Documents property templates

Document classes

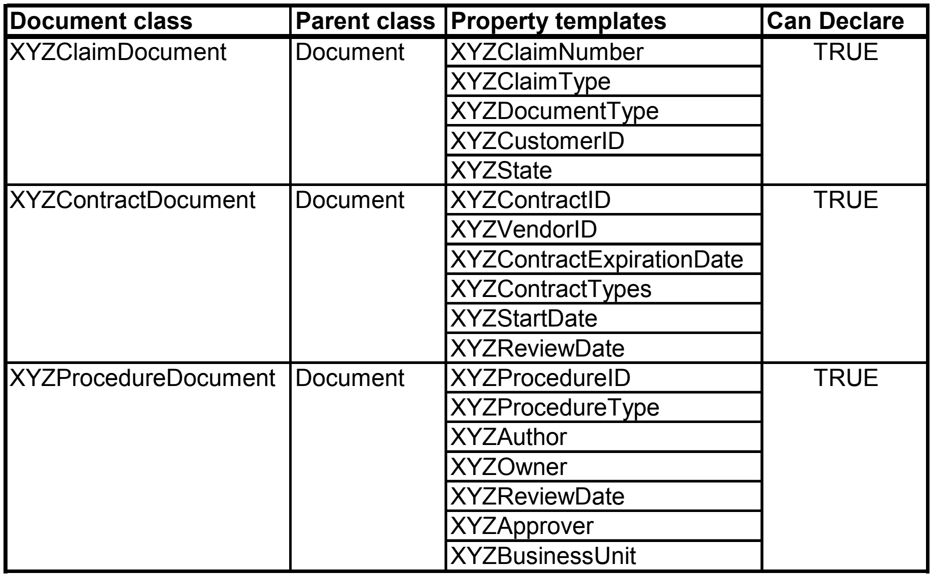
For the case study, these document classes are required to store contracts, claims, and procedure documents.

Because content belonging to these document classes can be declared as records, the class property *Can Declare* has to be set to TRUE for all these classes. It is possible to configure the object store so that all document classes can be declared as records by setting this value to TRUE on the root Document class and letting all subclasses inherit this property, which is *not* the best practice. If you enable this property at the root Document class level, all subclasses can potentially be declared as records, and there are subclasses that you do not want to be able to declare as records, for example:

- Publish Template
- Entry Template
- Stored Search
- Workflow Definition

We recommend to only enable classes that will be used to store content that will be declared as records.

Figure 11-18 on page 275 shows the document classes that we create for the RedBook_Documents object store.



Document class	Parent class	Property templates	Can Declare
XYZClaimDocument	Document	XYZClaimNumber	TRUE
		XYZClaimType	
		XYZDocumentType	
		XYZCustomerID	
		XYZState	
XYZContractDocument	Document	XYZContractID	TRUE
		XYZVendorID	
		XYZContractExpirationDate	
		XYZContractTypes	
		XYZStartDate	
		XYZReviewDate	
XYZProcedureDocument	Document	XYZProcedureID	TRUE
		XYZProcedureType	
		XYZAuthor	
		XYZOwner	
		XYZReviewDate	
		XYZApprover	
		XYZBusinessUnit	

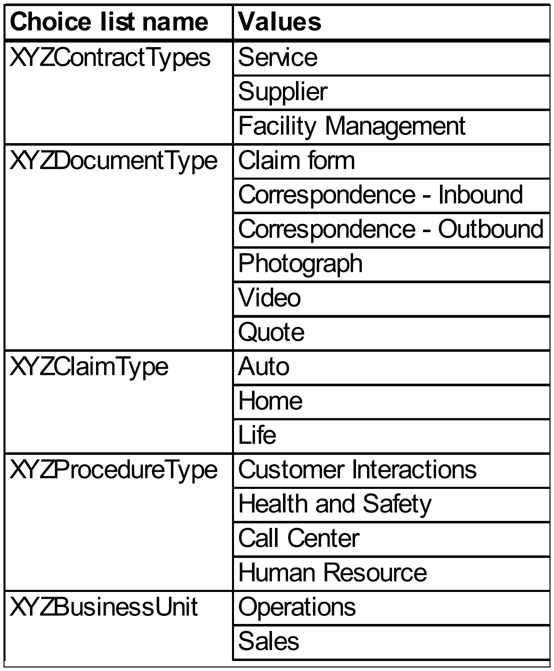
Figure 11-18 RedBook_Documents document classes

11.3.2 RedBook_Records object store (FPOS) setup

The RedBook_Records object store (FPOS) contains the file plan and the records metadata of the content (in ROS) that has been declared as records. We need to add choice lists, property templates, and document classes, including custom objects, for our case study.

Choice lists

Figure 11-19 on page 276 shows the choice lists that we create for the RedBook_Records object store. The choice lists are same as the ROS choice lists.

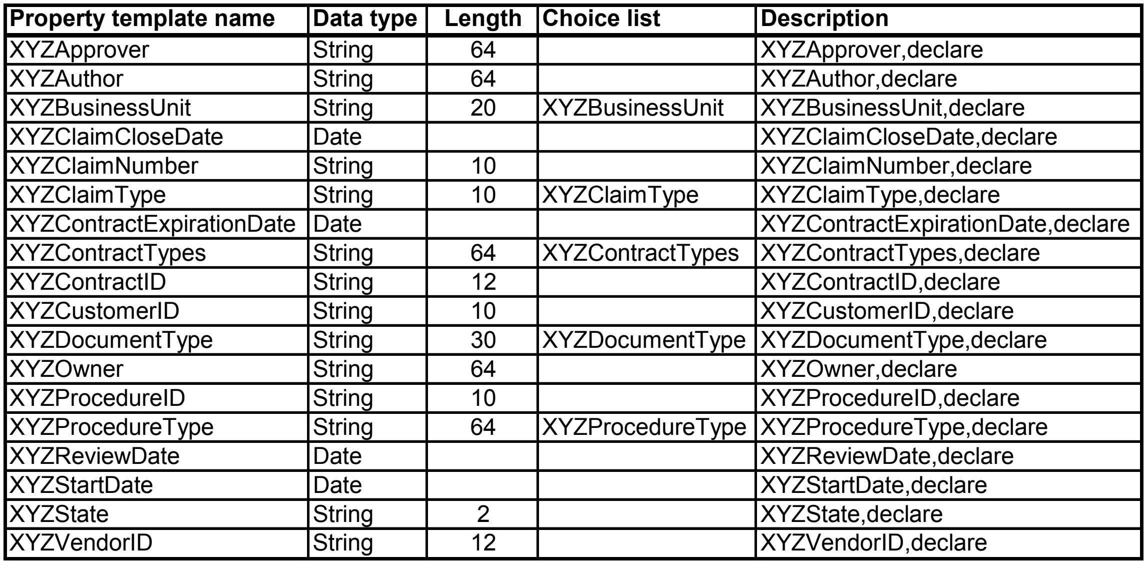


Choice list name	Values
XYZContractTypes	Service
	Supplier
	Facility Management
XYZDocumentType	Claim form
	Correspondence - Inbound
	Correspondence - Outbound
	Photograph
	Video
	Quote
XYZClaimType	Auto
	Home
	Life
XYZProcedureType	Customer Interactions
	Health and Safety
	Call Center
	Human Resource
XYZBusinessUnit	Operations
	Sales

Figure 11-19 RedBook_Records choice lists

Property templates

Figure 11-20 on page 277 shows the property templates that we need to create for the RedBook_Records object store.



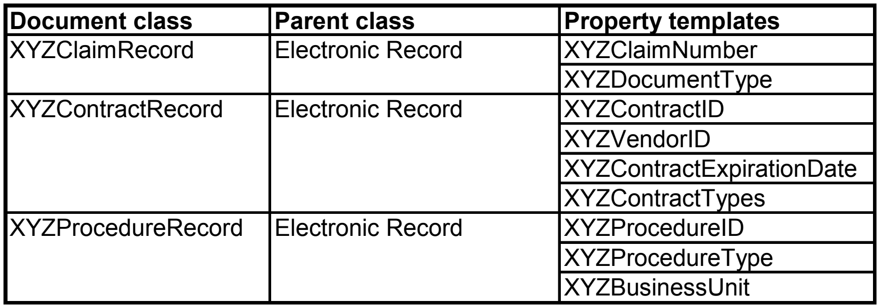
Property template name	Data type	Length	Choice list	Description
XYZApprover	String	64		XYZApprover,declare
XYZAuthor	String	64		XYZAuthor,declare
XYZBusinessUnit	String	20	XYZBusinessUnit	XYZBusinessUnit,declare
XYZClaimCloseDate	Date			XYZClaimCloseDate,declare
XYZClaimNumber	String	10		XYZClaimNumber,declare
XYZClaimType	String	10	XYZClaimType	XYZClaimType,declare
XYZContractExpirationDate	Date			XYZContractExpirationDate,declare
XYZContractTypes	String	64	XYZContractTypes	XYZContractTypes,declare
XYZContractID	String	12		XYZContractID,declare
XYZCustomerID	String	10		XYZCustomerID,declare
XYZDocumentType	String	30	XYZDocumentType	XYZDocumentType,declare
XYZOwner	String	64		XYZOwner,declare
XYZProcedureID	String	10		XYZProcedureID,declare
XYZProcedureType	String	64	XYZProcedureType	XYZProcedureType,declare
XYZReviewDate	Date			XYZReviewDate,declare
XYZStartDate	Date			XYZStartDate,declare
XYZState	String	2		XYZState,declare
XYZVendorID	String	12		XYZVendorID,declare

Figure 11-20 RedBook_Records property templates

> **Important:** The property templates in FPOS are similar to those property templates in ROS, except the property templates in FPOS must contain the word *declare* in the description fields. This word enables the automatic transferring of metadata from a declared content in ROS to the corresponding record object in FPOS.

Document classes

These document classes in FPOS hold the record objects for the corresponding document classes in the ROS. They are listed in Figure 11-21.



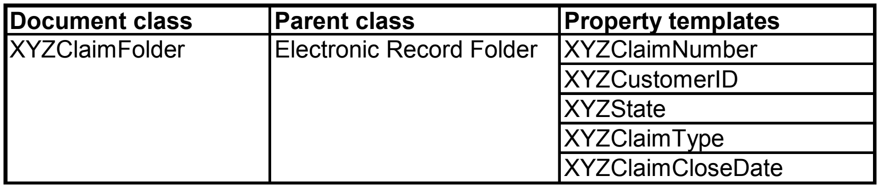
Document class	Parent class	Property templates
XYZClaimRecord	Electronic Record	XYZClaimNumber
		XYZDocumentType
XYZContractRecord	Electronic Record	XYZContractID
		XYZVendorID
		XYZContractExpirationDate
		XYZContractTypes
XYZProcedureRecord	Electronic Record	XYZProcedureID
		XYZProcedureType
		XYZBusinessUnit

Figure 11-21 RedBook_Records document classes

> **Note:** The property templates used for a document class in FPOS does not have to be the same property templates that are used for the corresponding ROS document class, because FPOS does not need to carry all the metadata of the content. It only needs data that facilitates the records management processes. In addition, you might have additional property templates in FPOS that are not originally in ROS for the same reason.

Custom objects

For the case study, we model the concept of a claim folder within IBM FileNet Records Manager. All claim documents are declared into the correct claim folder. Disposition aggregation is at the folder level. The folder that we define here is an Electronic Records Folder, with extra metadata. Create the new XYZClaimFolder in **Other Classes** → **Folder** → **RM Folder** → **Record Folder** → **Electronic Record Folder**. Refer to Figure 11-22 for its setup.



Document class	Parent class	Property templates
XYZClaimFolder	Electronic Record Folder	XYZClaimNumber
		XYZCustomerID
		XYZState
		XYZClaimType
		XYZClaimCloseDate

Figure 11-22 RedBook_Records custom objects

11.4 Configuring Workplace site preferences

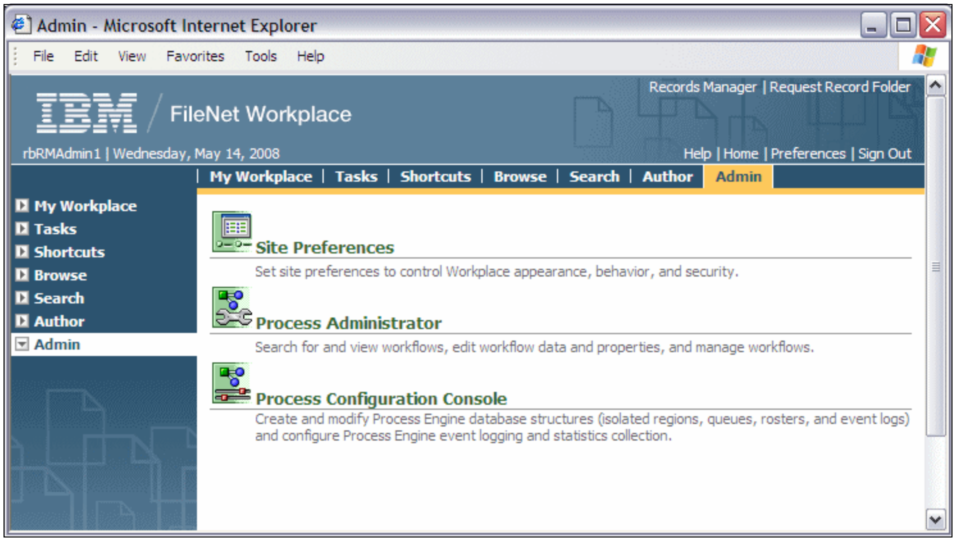
After preparing the ROS and FPOS object stores for records management activities and building the case study-specific data model objects, we need to set Workplace site preferences for these object stores:

- Setting the ROS to support records declaration.
- Setting the FPOS to support the file plan.

11.4.1 Setting the ROS to support records declaration

In Workplace site preferences, set ROS to support records declaration by following these steps:

1. Log on on to Workplace with an administrative ID.
2. Select the **Admin** tab. Figure 11-23 on page 279 appears.


Admin - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Records Manager | Request Record Folder
IBM FileNet Workplace
rbRMAdmin1 | Wednesday, May 14, 2008
Help | Home | Preferences | Sign Out
My Workplace | Tasks | Shortcuts | Browse | Search | Author | Admin
My Workplace
Tasks
Shortcuts
Browse
Search
Author
Admin

Site Preferences

Set site preferences to control Workplace appearance, behavior, and security.

Process Administrator

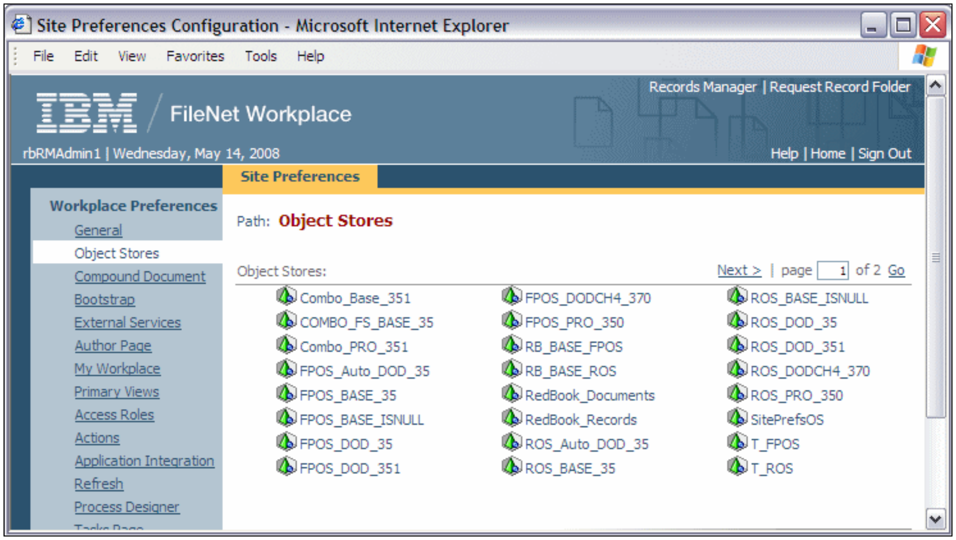
Search for and view workflows, edit workflow data and properties, and manage workflows.

Process Configuration Console

Create and modify Process Engine database structures (isolated regions, queues, rosters, and event logs) and configure Process Engine event logging and statistics collection.

Figure 11-23 Workplace: Admin tab

3. Click **Site Preferences**.
4. Click **Object Stores** from the left link bar, and a list of existing object stores appears on the right pane. Refer to Figure 11-24.



Path: **Object Stores**

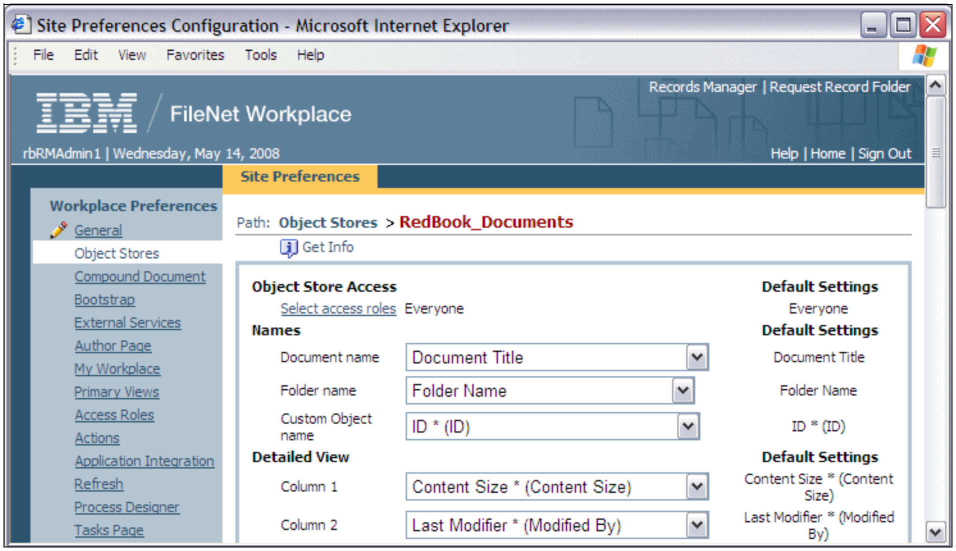
Object Stores:

Next > | page 1 of 2 Go

Combo_Base_351
COMBO_FS_BASE_35
Combo_PRO_351
FPOS_Auto_DOD_35
FPOS_BASE_35
FPOS_BASE_ISNULL
FPOS_DOD_35
FPOS_DOD_351
FPOS_DODCH4_370
FPOS_PRO_350
RB_BASE_FPOS
RB_BASE_ROS
RedBook_Documents
RedBook_Records
ROS_Auto_DOD_35
ROS_BASE_35
ROS_BASE_ISNULL
ROS_DOD_35
ROS_DOD_351
ROS_DODCH4_370
ROS_PRO_350
SitePrefsOS
T_FPOS
T_ROS

Figure 11-24 Site Preferences: Objects Stores

5. Select your ROS. For our case study, it is RedBook_Documents. Figure 11-25 appears.

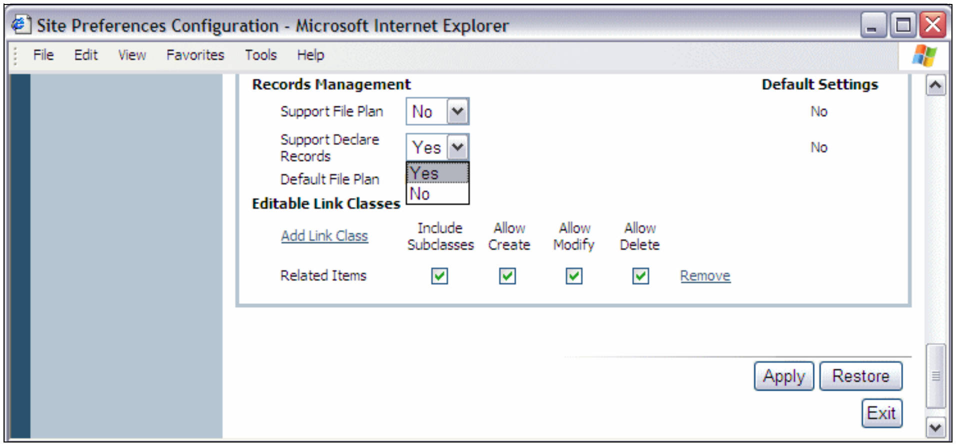


Path: Object Stores > RedBook_Documents

		Default Settings
Object Store Access		
Select access roles	Everyone	Everyone
Names		**Default Settings**
Document name	Document Title	Document Title
Folder name	Folder Name	Folder Name
Custom Object name	ID * (ID)	ID * (ID)
Detailed View		**Default Settings**
Column 1	Content Size * (Content Size)	Content Size * (Content Size)
Column 2	Last Modifier * (Modified By)	Last Modifier * (Modified By)

Figure 11-25 Site Preferences → Redbook_Documents (ROS)

6. Scroll down to the bottom of the Site Preferences page and set Support Declare Records to **Yes**. Refer to Figure 11-26.


Site Preferences Configuration - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Yes
No
Add Link Class
Remove
Apply
Restore
Exit

Records Management		**Default Settings**
Support File Plan	No	No
Support Declare Records	Yes	No
Default File Plan		

Editable Link Classes

	Include Subclasses	Allow Create	Allow Modify	Allow Delete	
Related Items	☑	☑	☑	☑	Remove

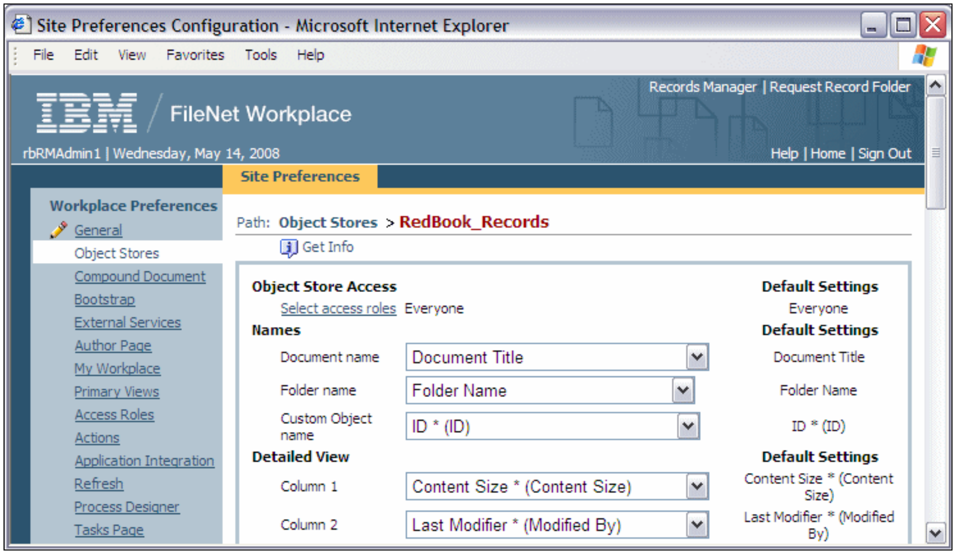
Figure 11-26 Site Preferences → RedBook_Documents (ROS) Support Declare Records

7. Click **Apply**, and then, click **Exit**.

11.4.2 Setting the FPOS to support the file plan

In Workplace site preferences, set the FPOS to support the file plan by following these steps:

1. From the Workplace, go to **Admin** → **Site Preferences** → **Object Stores** and select the FPOS. For the case study, it is RedBook_Records. Figure 11-27 appears.



Object Store Access | **Default Settings**
Select access roles: Everyone | Everyone

Names | **Default Settings**

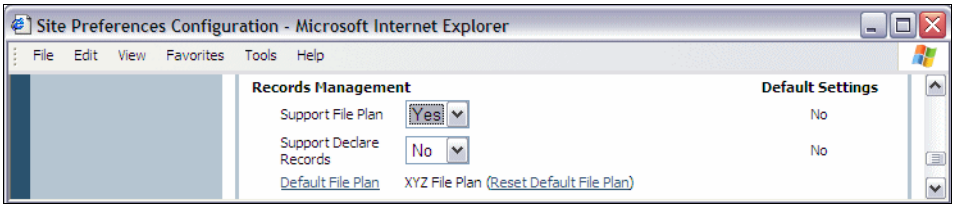
		Default Settings
Document name	Document Title	Document Title
Folder name	Folder Name	Folder Name
Custom Object name	ID * (ID)	ID * (ID)

Detailed View | **Default Settings**

		Default Settings
Column 1	Content Size * (Content Size)	Content Size * (Content Size)
Column 2	Last Modifier * (Modified By)	Last Modifier * (Modified By)

Figure 11-27 Site Preferences → Redbook_Records (FPOS)

2. Scroll down to the bottom of the page to set Support File Plan to **Yes**. Refer to Figure 11-28.


Site Preferences Configuration - Microsoft Internet Explorer
File Edit View Favorites Tools Help

Records Management | **Default Settings**

		Default Settings
Support File Plan	Yes	No
Support Declare Records	No	No
Default File Plan	XYZ File Plan (Reset Default File Plan)	

Figure 11-28 Site Preferences → RedBook_Records (FPOS) Support File Plan

3. Click **Apply,** and then, click **Exit**.

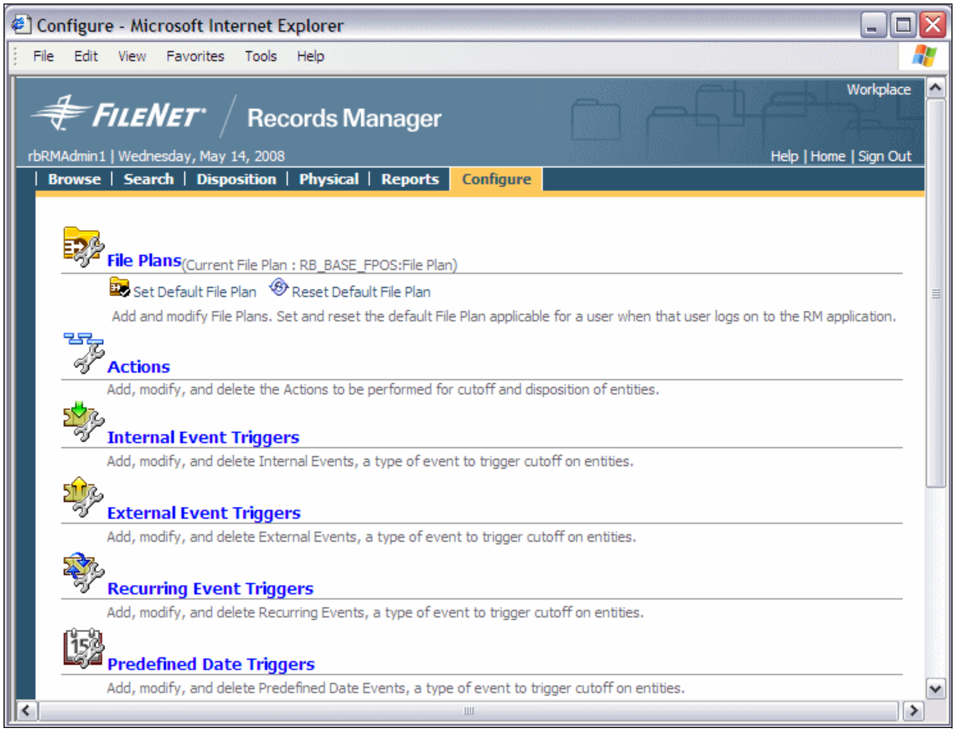
11.5 Creating a file plan

The file plan object has been enabled for file plan support, so we can now create a file plan. Follow these steps:

1. Launch the Records Manager Web application:

   http://<*Host name*>:8080/RecordsManager

2. Log in with a user ID that is a member of the Records Administrator role.
3. Select the **Configure** tab. Refer to Figure 11-29.


Configure - Microsoft Internet Explorer
File Edit View Favorites Tools Help
FILENET Records Manager
Workplace
rbRMAdmin1 | Wednesday, May 14, 2008
Help | Home | Sign Out
Browse | Search | Disposition | Physical | Reports | Configure

File Plans (Current File Plan : RB_BASE_FPOS:File Plan)

Set Default File Plan   Reset Default File Plan

Add and modify File Plans. Set and reset the default File Plan applicable for a user when that user logs on to the RM application.

Actions

Add, modify, and delete the Actions to be performed for cutoff and disposition of entities.

Internal Event Triggers

Add, modify, and delete Internal Events, a type of event to trigger cutoff on entities.

External Event Triggers

Add, modify, and delete External Events, a type of event to trigger cutoff on entities.

Recurring Event Triggers

Add, modify, and delete Recurring Events, a type of event to trigger cutoff on entities.

Predefined Date Triggers

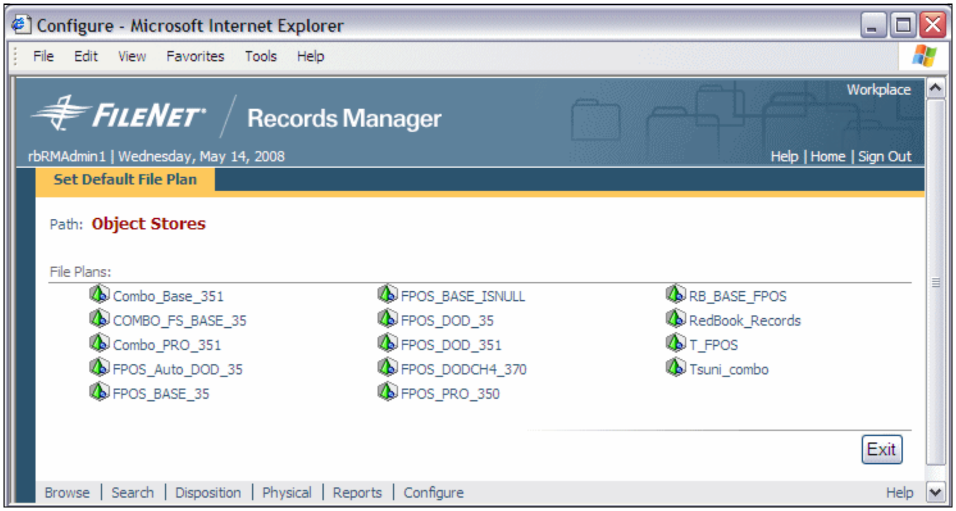
Add, modify, and delete Predefined Date Events, a type of event to trigger cutoff on entities.

Figure 11-29 Records Manager application: Configure tab

4. Set the default file plan.

   In order to create a new file plan in the new FPOS, you need to specify a default file plan. The file plan named *File Plan*, which is created by default in an FPOS, needs to be defined as the default file plan. You might also have multiple FPOSs in your IBM FileNet P8 domain, but not likely. To select the default file plan:

   a. Click **Set Default File Plan**. Refer to Figure 11-30 on page 283.


Configure - Microsoft Internet Explorer
File Edit View Favorites Tools Help
FILENET Records Manager
Workplace
rbRMAdmin1 | Wednesday, May 14, 2008
Help | Home | Sign Out
Set Default File Plan
Exit
Browse | Search | Disposition | Physical | Reports | Configure
Help

Path: Object Stores

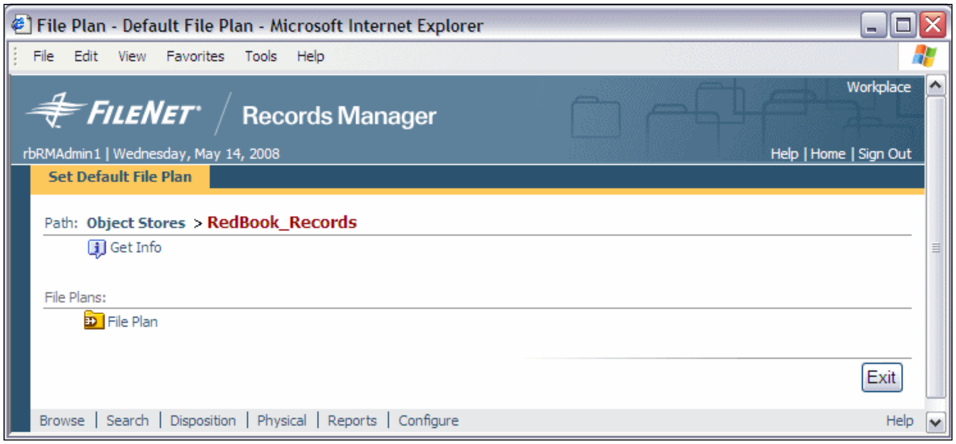
File Plans:

Combo_Base_351
COMBO_FS_BASE_35
Combo_PRO_351
FPOS_Auto_DOD_35
FPOS_BASE_35
FPOS_BASE_ISNULL
FPOS_DOD_35
FPOS_DOD_351
FPOS_DODCH4_370
FPOS_PRO_350
RB_BASE_FPOS
RedBook_Records
T_FPOS
Tsuni_combo

Figure 11-30 Configure → Set Default File Plan: Select object store

b. Select your FPOS. For our case study, it is RedBook_Records.

c. Select **File Plan**. Refer to Figure 11-31.


File Plan - Default File Plan - Microsoft Internet Explorer
File Edit View Favorites Tools Help
FILENET Records Manager
Workplace
rbRMAdmin1 | Wednesday, May 14, 2008
Help | Home | Sign Out
Set Default File Plan
Get Info
Exit
Browse | Search | Disposition | Physical | Reports | Configure
Help

Path: Object Stores > RedBook_Records

File Plans:

File Plan

Figure 11-31 Configure → Redbook_Records: Select File Plan as the default file plan

d. Click **Accept**. Refer to Figure 11-32 on page 284.

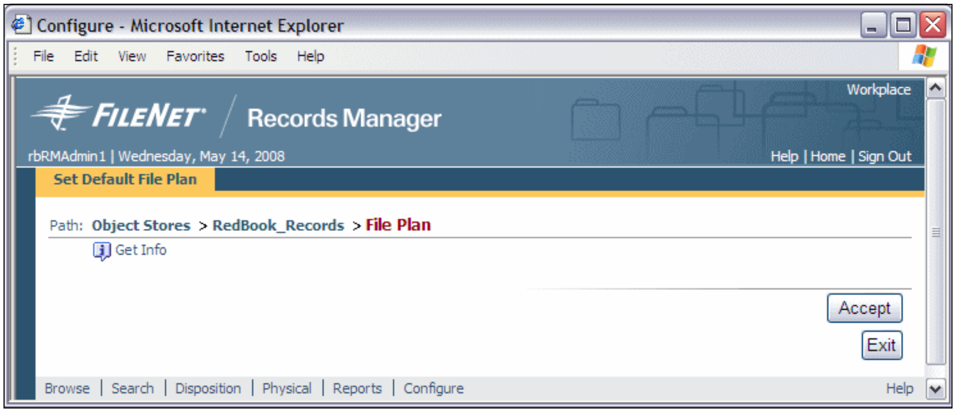
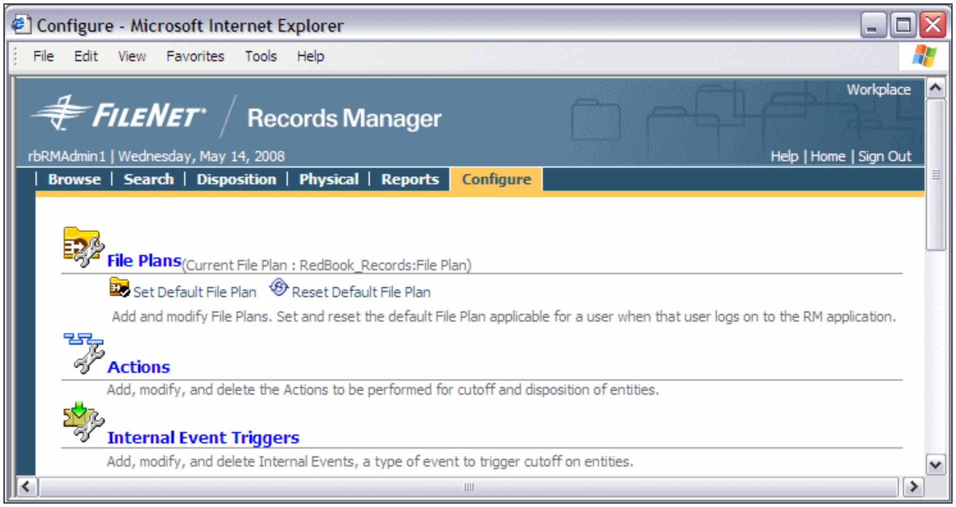

Configure - Microsoft Internet Explorer
File Edit View Favorites Tools Help
FILENET Records Manager
Workplace
rbRMAdmin1 | Wednesday, May 14, 2008
Help | Home | Sign Out
Set Default File Plan
Path: Object Stores > RedBook_Records > File Plan
Get Info
Accept
Exit
Browse | Search | Disposition | Physical | Reports | Configure
Help

Figure 11-32 Configure → Redbook_Records: Accept File Plan as the default file plan

e. Select the **Configure** tab again and you now see that the default file plan is set in your FPOS (RedBook_Records). Refer to Figure 11-33.


Configure - Microsoft Internet Explorer
File Edit View Favorites Tools Help
FILENET Records Manager
Workplace
rbRMAdmin1 | Wednesday, May 14, 2008
Help | Home | Sign Out
| Browse | Search | Disposition | Physical | Reports | Configure
File Plans(Current File Plan : RedBook_Records:File Plan)
Set Default File Plan
Reset Default File Plan

Add and modify File Plans. Set and reset the default File Plan applicable for a user when that user logs on to the RM application.

Add, modify, and delete the Actions to be performed for cutoff and disposition of entities.

Add, modify, and delete Internal Events, a type of event to trigger cutoff on entities.

Figure 11-33 Configure → Current File Plan is set to Redbook_Records' File Plan

5. Create a new file plan.

   At this point, you have two options for creating your file plan. You can either:

   – Rename the existing file plan (File Plan). The best practice in a production system is to only have a single file plan in an FPOS.

   – Create a new file plan, which is what we did for our case study, because we want to try configuration options.

To create a new file plan:

a. Select **File Plans** from the Configure tab.

b. Select **Add File Plan**. Refer to Figure 11-34.

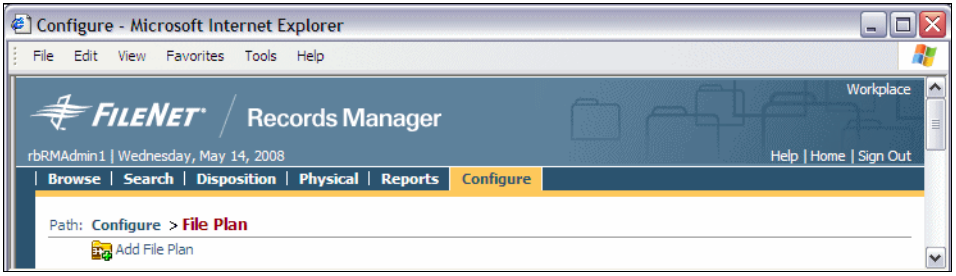
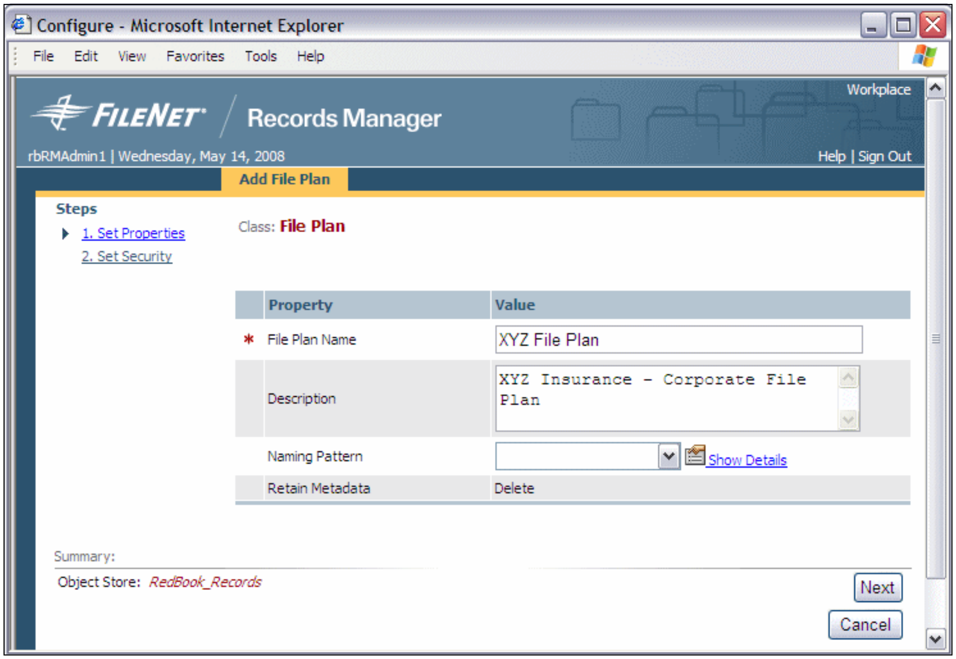

Configure - Microsoft Internet Explorer
File Edit View Favorites Tools Help
FILENET Records Manager
Workplace
rbRMAdmin1 | Wednesday, May 14, 2008
Help | Home | Sign Out
Browse | Search | Disposition | Physical | Reports | Configure
Path: Configure > File Plan
Add File Plan

Figure 11-34 Configure → File Plan: Add File Plan

c. Enter information for File Plan Name and Description fields (and optionally a Naming Pattern[1]). For our case study, we enter XYZ File Plan as the new file plan name. Refer to Figure 11-35.



Class: File Plan

	Property	Value
*	File Plan Name	XYZ File Plan
	Description	XYZ Insurance - Corporate File Plan
	Naming Pattern	
	Retain Metadata	Delete

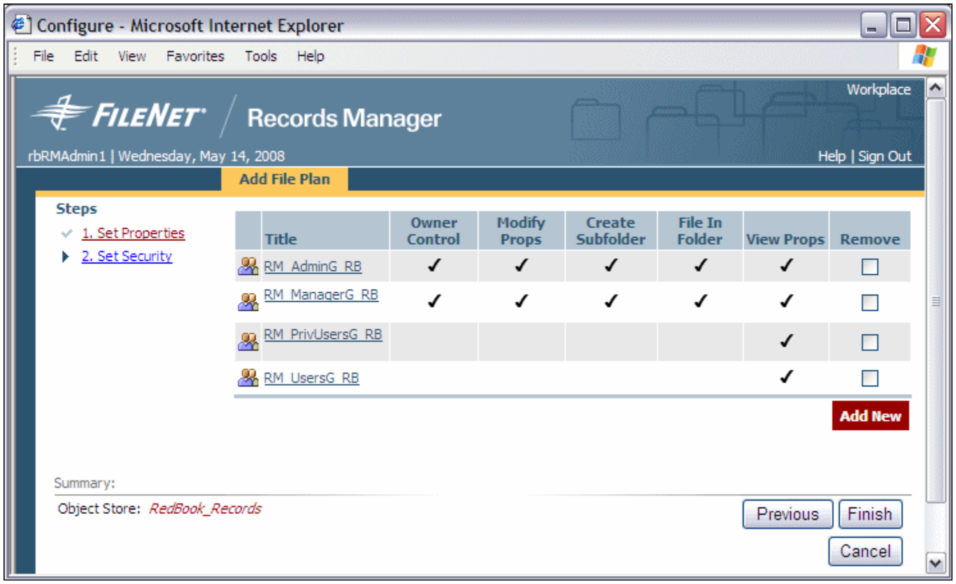
Object Store: RedBook_Records

Figure 11-35 Configure → Add XYZ File Plan: Set properties

[1] Naming Patterns are a mechanism to ensure consistency in naming to the record category names and IDs, but we do not use them in this book.

d. Click **Next**.

e. Set the security for the file plan. For the case study, we use the default. Refer to Figure 11-36.



rbRMAdmin1 | Wednesday, May 14, 2008

Steps

- 1. Set Properties
- 2. Set Security

Title	Owner Control	Modify Props	Create Subfolder	File In Folder	View Props	Remove
RM_AdminG_RB	✓	✓	✓	✓	✓	□
RM_ManagerG_RB	✓	✓	✓	✓	✓	□
RM_PrivUsersG_RB					✓	□
RM_UsersG_RB					✓	□

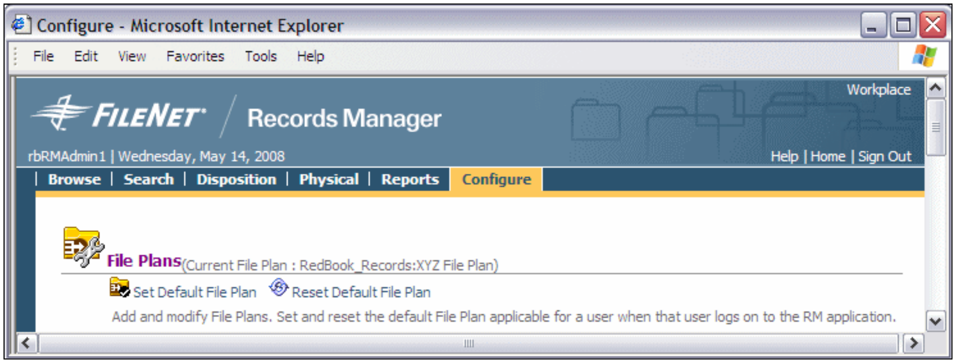
Summary:

Object Store: *RedBook_Records*

Figure 11-36 Configure → Add XYZ File Plan: Set Security

f. Click **Finish**, and then click **OK** on the window that says that the file plan has been successfully created.

6. Repeat step 4, “Set the correct default file plan.” Except this time, select the new file plan that you have just created as the default file plan. For our case study, we select **XYZ File Plan** as our default file plan. Refer to Figure 11-37.


Configure - Microsoft Internet Explorer
File Edit View Favorites Tools Help
FileNet Records Manager
Workplace
Help | Home | Sign Out
Browse | Search | Disposition | Physical | Reports | Configure
Set Default File Plan
Reset Default File Plan

rbRMAdmin1 | Wednesday, May 14, 2008

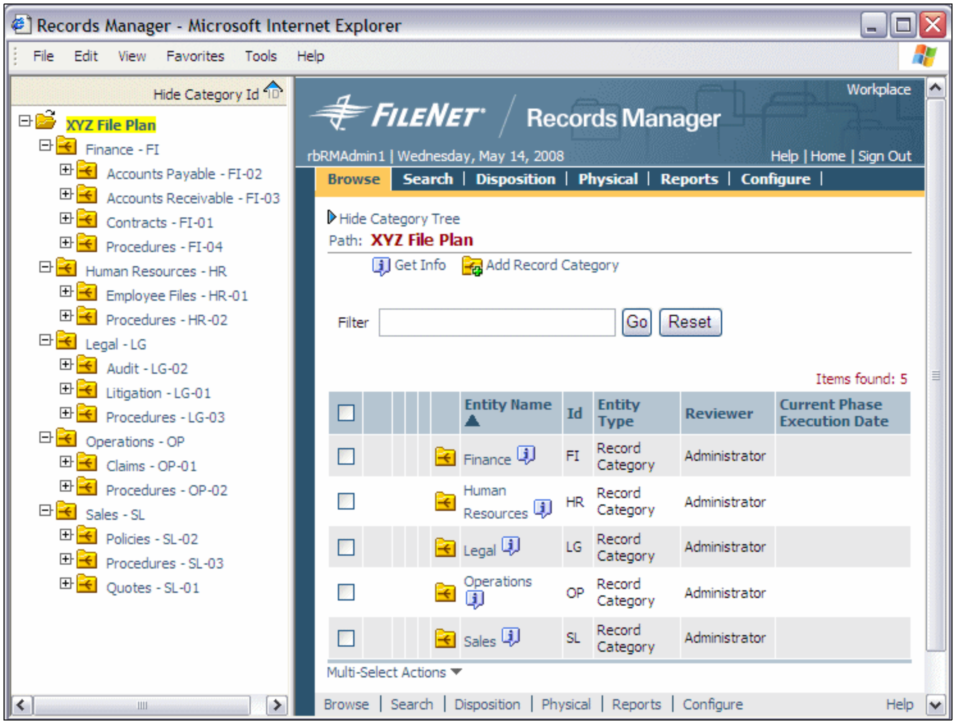
File Plans(Current File Plan : RedBook_Records:XYZ File Plan)

Add and modify File Plans. Set and reset the default File Plan applicable for a user when that user logs on to the RM application.

Figure 11-37 Configure → File Plans: Current File Plan set to XYZ File Plan

11.5.1 Build the file plan categories

Having created the file plan, we now need to build the file plan categories. We create a hierarchy of record categories that maps to Figure 11-1 on page 256. Figure 11-38 shows the completed file plan for our case study.


Records Manager - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Hide Category Id
FILENET Records Manager
Workplace
rbRMAdmin1 | Wednesday, May 14, 2008
Help | Home | Sign Out
Browse | Search | Disposition | Physical | Reports | Configure |
Hide Category Tree
Get Info
Add Record Category
Filter
Go
Reset
Multi-Select Actions
Browse | Search | Disposition | Physical | Reports | Configure
Help

- XYZ File Plan
  - Finance - FI
    - Accounts Payable - FI-02
    - Accounts Receivable - FI-03
    - Contracts - FI-01
    - Procedures - FI-04
  - Human Resources - HR
    - Employee Files - HR-01
    - Procedures - HR-02
  - Legal - LG
    - Audit - LG-02
    - Litigation - LG-01
    - Procedures - LG-03
  - Operations - OP
    - Claims - OP-01
    - Procedures - OP-02
  - Sales - SL
    - Policies - SL-02
    - Procedures - SL-03
    - Quotes - SL-01

Path: XYZ File Plan

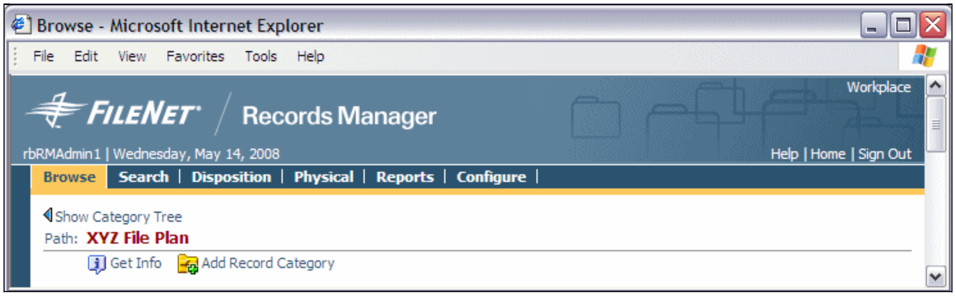
Items found: 5

□	Entity Name	Id	Entity Type	Reviewer	Current Phase Execution Date
□	Finance	FI	Record Category	Administrator	
□	Human Resources	HR	Record Category	Administrator	
□	Legal	LG	Record Category	Administrator	
□	Operations	OP	Record Category	Administrator	
□	Sales	SL	Record Category	Administrator	

Figure 11-38 Browse → XYZ File Plan expanded

To create a record category:

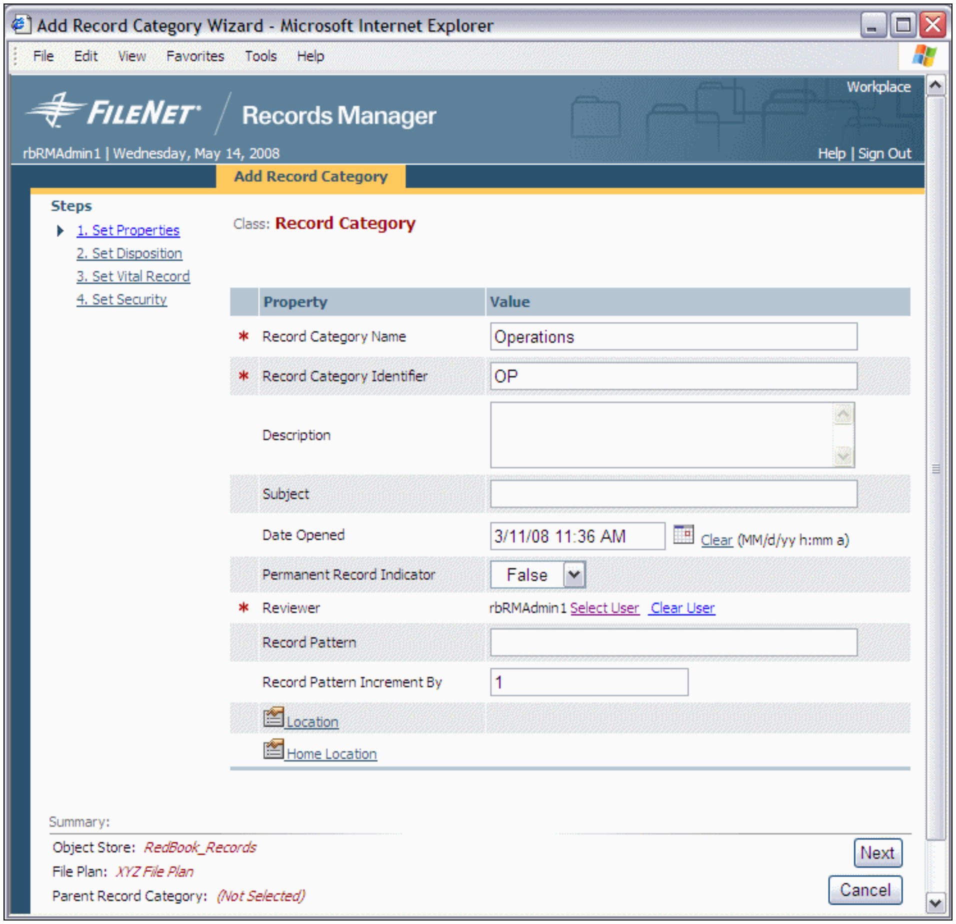
1. From the IBM FileNet Records Manager Web application, select **Browse**.
2. Make sure that you are working with your current file plan. Select **Add Record Category**. Refer to Figure 11-39 on page 288.


Browse - Microsoft Internet Explorer
File Edit View Favorites Tools Help
FILENET Records Manager
Workplace
rbRMAdmin1 | Wednesday, May 14, 2008
Help | Home | Sign Out
Browse | Search | Disposition | Physical | Reports | Configure |
Show Category Tree
Get Info
Add Record Category

Path: XYZ File Plan

Figure 11-39 XYZ File Plan Add Record Category

3. Enter the record category information. Select **Reviewer** for the category. Figure 11-40 shows the setup for the Operations record category.



Class: Record Category

	Property	Value
*	Record Category Name	Operations
*	Record Category Identifier	OP
	Description	
	Subject	
	Date Opened	3/11/08 11:36 AM Clear (MM/d/yy h:mm a)
	Permanent Record Indicator	False
*	Reviewer	rbRMAdmin1 Select User Clear User
	Record Pattern	
	Record Pattern Increment By	1
	Location	
	Home Location	

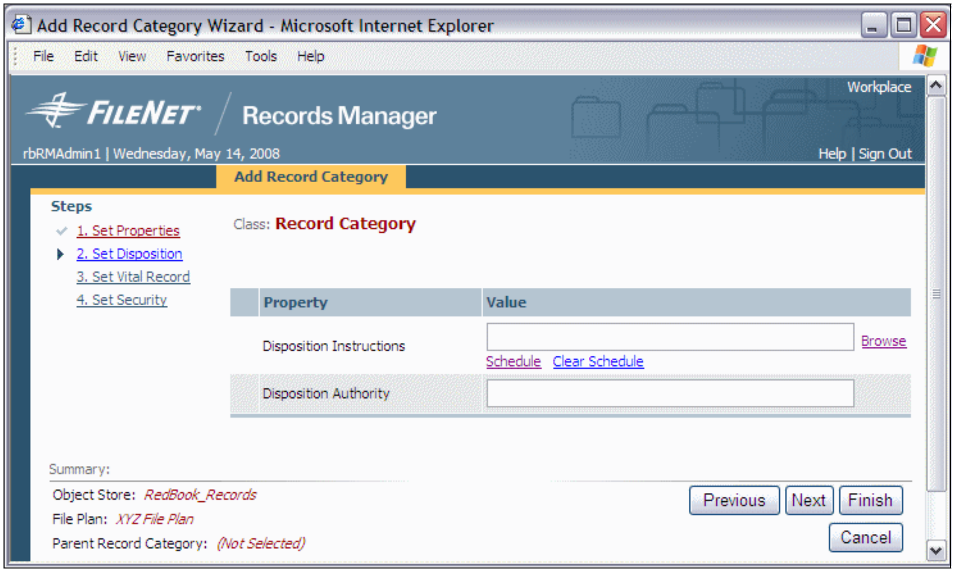
Summary:
Object Store: *RedBook_Records*
File Plan: *XYZ File Plan*
Parent Record Category: *(Not Selected)*

Figure 11-40 XYZ File Plan Add Record Category (Operations): Set Properties

4. Click **Next**.

5. Assign an appropriate disposition schedule for this category. If there is no associated disposition schedule or you have not defined it yet, click **Next** to continue. To assign a disposition schedule:

   a. At the Set Disposition window (Figure 11-41), click **Schedule**.



rbRMAdmin1 | Wednesday, May 14, 2008

Class: Record Category

Property	Value
Disposition Instructions	
Disposition Authority	

Summary:

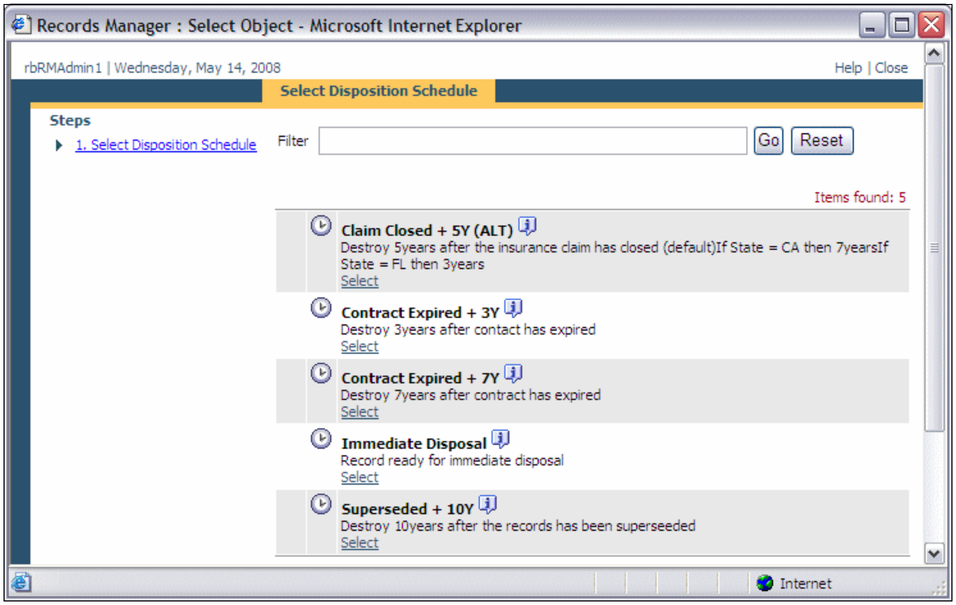
Object Store: *RedBook_Records*

File Plan: *XYZ File Plan*

Parent Record Category: *(Not Selected)*

Figure 11-41 XYZ File Plan Add Record Category (Operations): Set Disposition

   b. Click **Select** under the disposition schedule that you want to assign to the record category. Figure 11-42 on page 290 shows a list of the disposition schedules that we have already created for our case study.


Records Manager : Select Object - Microsoft Internet Explorer
rbRMAdmin1 | Wednesday, May 14, 2008
Help | Close
Select Disposition Schedule
Steps
1. Select Disposition Schedule
Filter
Go
Reset
Items found: 5
Select
Internet

Claim Closed + 5Y (ALT)
Destroy 5years after the insurance claim has closed (default)If State = CA then 7yearsIf State = FL then 3years

Contract Expired + 3Y
Destroy 3years after contact has expired

Contract Expired + 7Y
Destroy 7years after contract has expired

Immediate Disposal
Record ready for immediate disposal

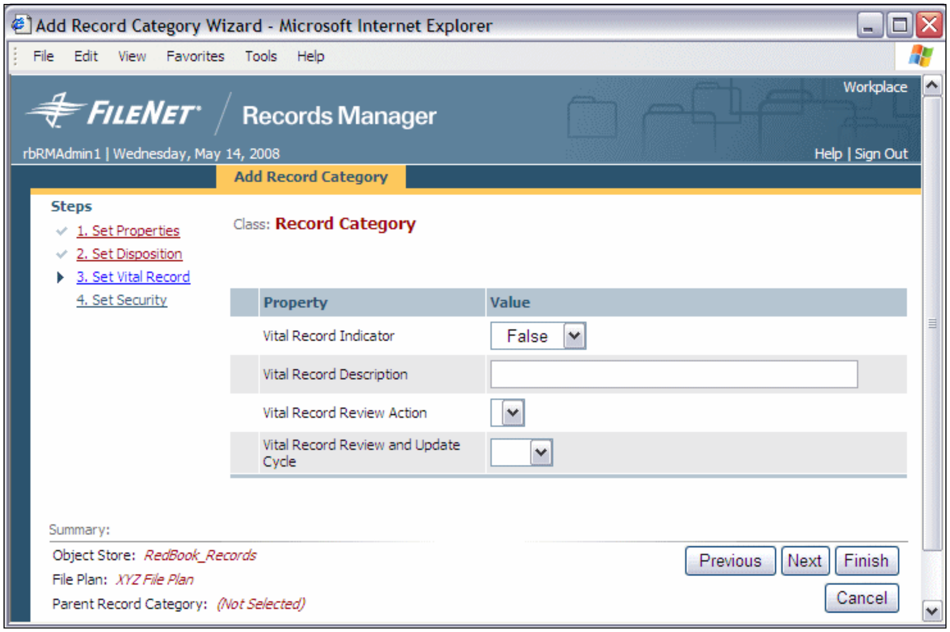
Superseded + 10Y
Destroy 10years after the records has been superseeded

Figure 11-42 XYZ File Plan Add Record Category: Select Disposition Schedule

c. Click **Next** on the Add Record Category Set Disposition Schedule window.

In later chapters, we show you how to create a disposition schedule and associate it to a file plan component.

6. Click **Set Vital Record** to update information if necessary. For our case study, we accept defaults.
7. Click **Next**. Refer to Figure 11-43 on page 291.



Class: **Record Category**

Property	Value
Vital Record Indicator	False
Vital Record Description	
Vital Record Review Action	
Vital Record Review and Update Cycle	

Summary:

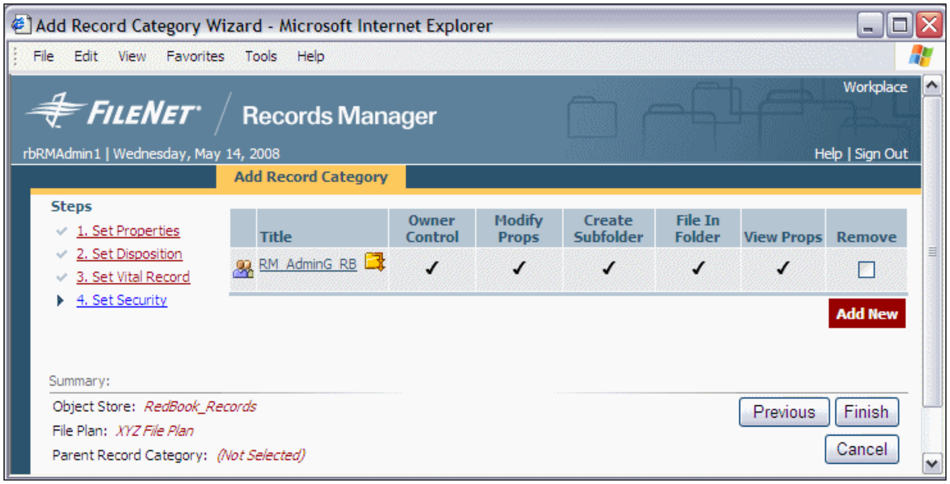
Object Store: *RedBook_Records*

File Plan: *XYZ File Plan*

Parent Record Category: *(Not Selected)*

Figure 11-43 XYZ File Plan Add Record Category (Operations): Set Vital Record

8. In the Set Security window, update the security details if necessary. Refer to Figure 11-44. By default, only members of the Records Administrator group RM_AdminG_RB can access this category.



Title	Owner Control	Modify Props	Create Subfolder	File In Folder	View Props	Remove
RM_AdminG_RB	✓	✓	✓	✓	✓	□

Summary:

Object Store: *RedBook_Records*

File Plan: *XYZ File Plan*

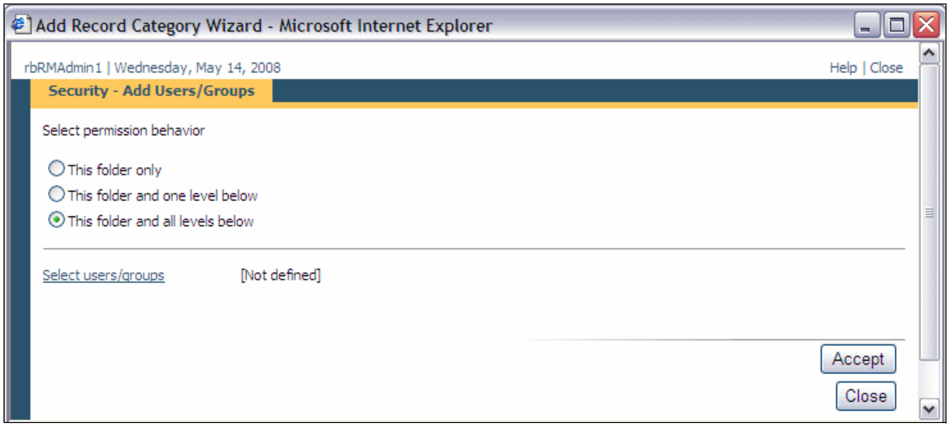
Parent Record Category: *(Not Selected)*

Figure 11-44 XYZ File Plan Add Record Category (Operations): Set Security

For our case study, we add the RM_OperationsG_RB organizational group so that they have access to their records. To add new users and groups to the security:

a. Click **Add New** from the previous window.

b. Select the appropriate permission behavior. For our case study, we want the security permission to be applicable for this folder and all levels beneath it. We select **This folder and all levels below**. Refer to Figure 11-45.



Add Record Category Wizard - Microsoft Internet Explorer

rbRMAdmin1 | Wednesday, May 14, 2008

Security - Add Users/Groups

Select permission behavior

○ This folder only
○ This folder and one level below
● This folder and all levels below

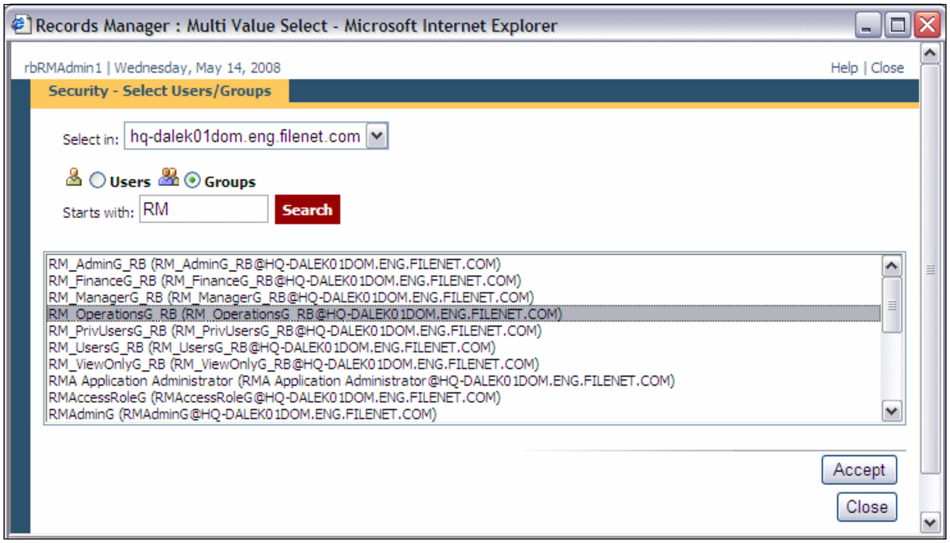
Select users/groups [Not defined]

Figure 11-45 XYZ File Plan Add Record Category: Set Security - Add Users/Groups

c. Click **Select users/groups**. Figure 11-46 on page 293 appears.

d. Select whether you are searching for users or groups. The best practice is to always work with groups.

e. Enter a search string in the “Starts with” text box and click **Search**. This string will be searched against LDAP for users and groups. For our case study, we enter RM.


Records Manager : Multi Value Select - Microsoft Internet Explorer

rbRMAdmin1 | Wednesday, May 14, 2008

Security - Select Users/Groups

Select in: hq-dalek01dom.eng.filenet.com

○ Users ● Groups

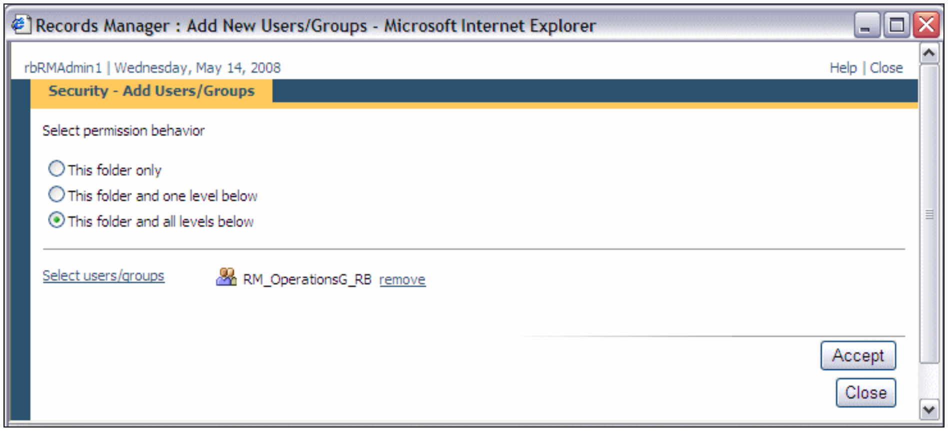
Starts with: RM

RM_AdminG_RB (RM_AdminG_RB@HQ-DALEK01DOM.ENG.FILENET.COM)
RM_FinanceG_RB (RM_FinanceG_RB@HQ-DALEK01DOM.ENG.FILENET.COM)
RM_ManagerG_RB (RM_ManagerG_RB@HQ-DALEK01DOM.ENG.FILENET.COM)
RM_OperationsG_RB (RM_OperationsG_RB@HQ-DALEK01DOM.ENG.FILENET.COM)
RM_PrivUsersG_RB (RM_PrivUsersG_RB@HQ-DALEK01DOM.ENG.FILENET.COM)
RM_UsersG_RB (RM_UsersG_RB@HQ-DALEK01DOM.ENG.FILENET.COM)
RM_ViewOnlyG_RB (RM_ViewOnlyG_RB@HQ-DALEK01DOM.ENG.FILENET.COM)
RMA Application Administrator (RMA Application Administrator@HQ-DALEK01DOM.ENG.FILENET.COM)
RMAccessRoleG (RMAccessRoleG@HQ-DALEK01DOM.ENG.FILENET.COM)
RMAdminG (RMAdminG@HQ-DALEK01DOM.ENG.FILENET.COM)

Figure 11-46 XYZ File Plan Add Record Category: Set Security - Select Users/Groups

f. From the search result list, select the appropriate group and click **Accept**. For our case study, we select **RM_OperationsG_RB**.

g. You are returned to the **Security - Add Users/Groups** window. Figure 11-47 shows that the new group information is added.


Records Manager : Add New Users/Groups - Microsoft Internet Explorer

rbRMAdmin1 | Wednesday, May 14, 2008

Security - Add Users/Groups

Select permission behavior

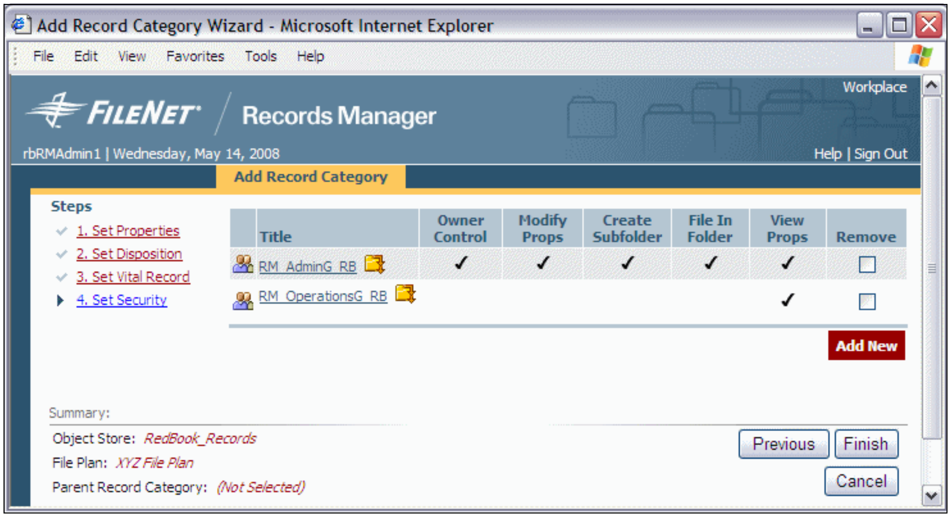
○ This folder only
○ This folder and one level below
● This folder and all levels below

Select users/groups RM_OperationsG_RB remove

Figure 11-47 XYZ File Plan Add Record Category: Security users/groups added

h. Click **Accept**.

The RM_OperationsG_RB group is added to the security profile for the category. Refer to Figure 11-48 on page 294.



rbRMAdmin1 | Wednesday, May 14, 2008

Title	Owner Control	Modify Props	Create Subfolder	File In Folder	View Props	Remove
RM_AdminG_RB	✓	✓	✓	✓	✓	□
RM_OperationsG_RB					✓	□

Summary:

Object Store: *RedBook_Records*

File Plan: *XYZ File Plan*

Parent Record Category: *(Not Selected)*

Figure 11-48 XYZ File Plan Add Record Category: Security updated

i. Click **Finish**, and then, click **OK**.

9. Repeat this process for additional record categories in the file plan. Note that if you do not have any disposition schedules set up at this point and do not need to update the Vital Record information or change the Security setup, you can click Finish after you have entered the record category name and identifier.

   To add the remaining record categories:

   a. Add the *level 1* records categories as listed in Figure 11-49.

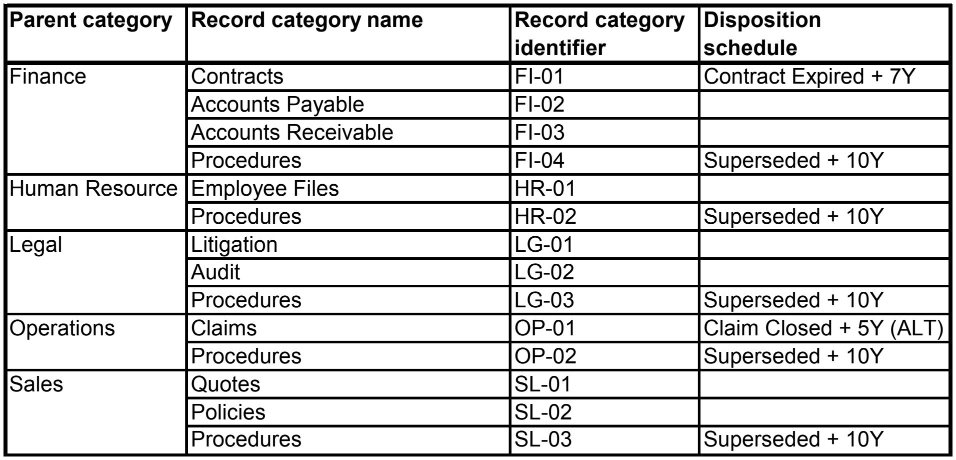
Parent category	Record category name	Record category identifier	Disposition schedule
	Finance	FI	
	Human Resouces	HR	
	Legal	LG	
	Operations	OP	

Figure 11-49 XYZ File Plan: Level 1 record categories

   b. Add the *level 2* record categories as listed in Figure 11-50 on page 295.

   To add the level 2 record categories, select the parent category first, and then, click **Add Records Category**, and enter the appropriate records category name and identifier. When done, click the root category file plan path (in our case study, it is XYZ File Plan) and you are back to the Level

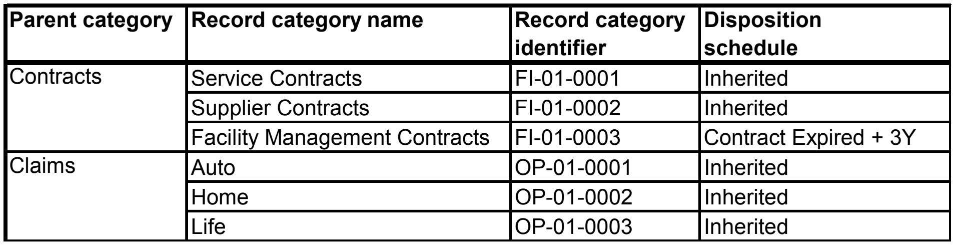
1 record categories. If you do not have any disposition schedules set up yet at this time, do not assign them.



Parent category	Record category name	Record category identifier	Disposition schedule
Finance	Contracts	FI-01	Contract Expired + 7Y
	Accounts Payable	FI-02	
	Accounts Receivable	FI-03	
	Procedures	FI-04	Superseded + 10Y
Human Resource	Employee Files	HR-01	
	Procedures	HR-02	Superseded + 10Y
Legal	Litigation	LG-01	
	Audit	LG-02	
	Procedures	LG-03	Superseded + 10Y
Operations	Claims	OP-01	Claim Closed + 5Y (ALT)
	Procedures	OP-02	Superseded + 10Y
Sales	Quotes	SL-01	
	Policies	SL-02	
	Procedures	SL-03	Superseded + 10Y

Figure 11-50 XYZ File Plan: Level 2 record categories

c. Add *level 3* record categories as listed in Figure 11-51. If you do not have the disposition schedules set up at this point, do not assign them.



Parent category	Record category name	Record category identifier	Disposition schedule
Contracts	Service Contracts	FI-01-0001	Inherited
	Supplier Contracts	FI-01-0002	Inherited
	Facility Management Contracts	FI-01-0003	Contract Expired + 3Y
Claims	Auto	OP-01-0001	Inherited
	Home	OP-01-0002	Inherited
	Life	OP-01-0003	Inherited

Figure 11-51 XYZ File Plan: Level 3 record categories

The file plan for the case study is now complete and ready for the Claim folder to be created and records to be declared into it. To see all the record categories that you created, click **Show Category Tree** and the system displays the categories in tree format.

11.5.2 Creating an electronic record folder

Typically, the electronic record folder (XYZClaimFolder), which we use to declare claim records into, is created by an automated mechanism triggered through a workflow or an external system. For the purpose of completeness within this case study, we document the manual creation process of this folder.

To create an electronic record folder:

1. Log on to IBM FileNet Records Manager Web application with an administrative ID.
2. Navigate through the file plan to the location where you want to create a folder. For our case study, go to **XYZ File Plan** → **Operations** → **Claims** → **Auto**. If the category tree does not display on the left pane, click **Show Category Tree**.
3. Select **Add Record Folder**. Refer to Figure 11-52.

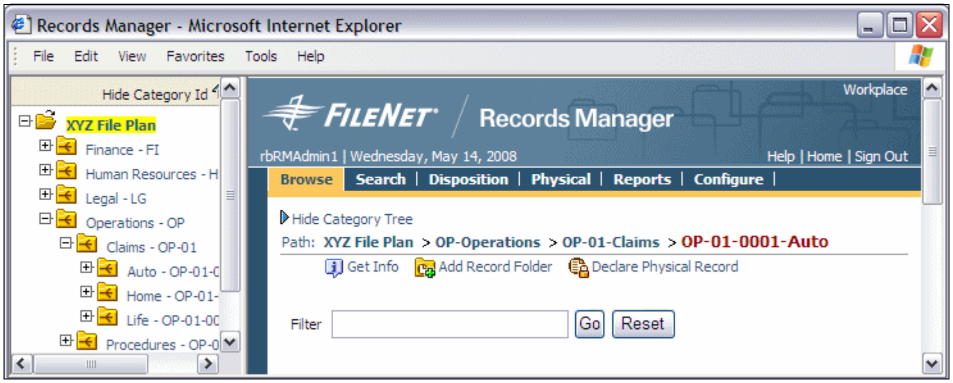
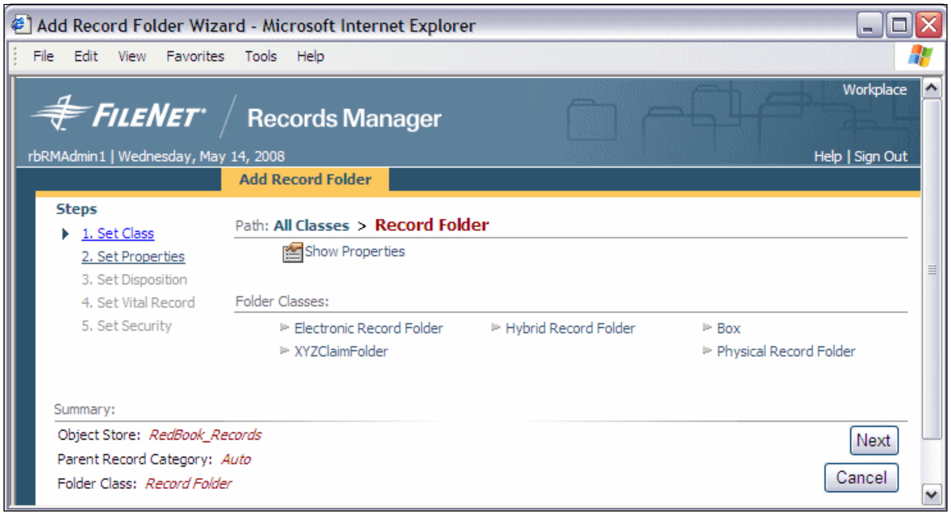


Figure 11-52 XYZ File Plan Add Record Folder

4. Choose the appropriate Record Folder class and click **Next**. For our case study, we select **XYZClaimFolder**. Refer to Figure 11-53.



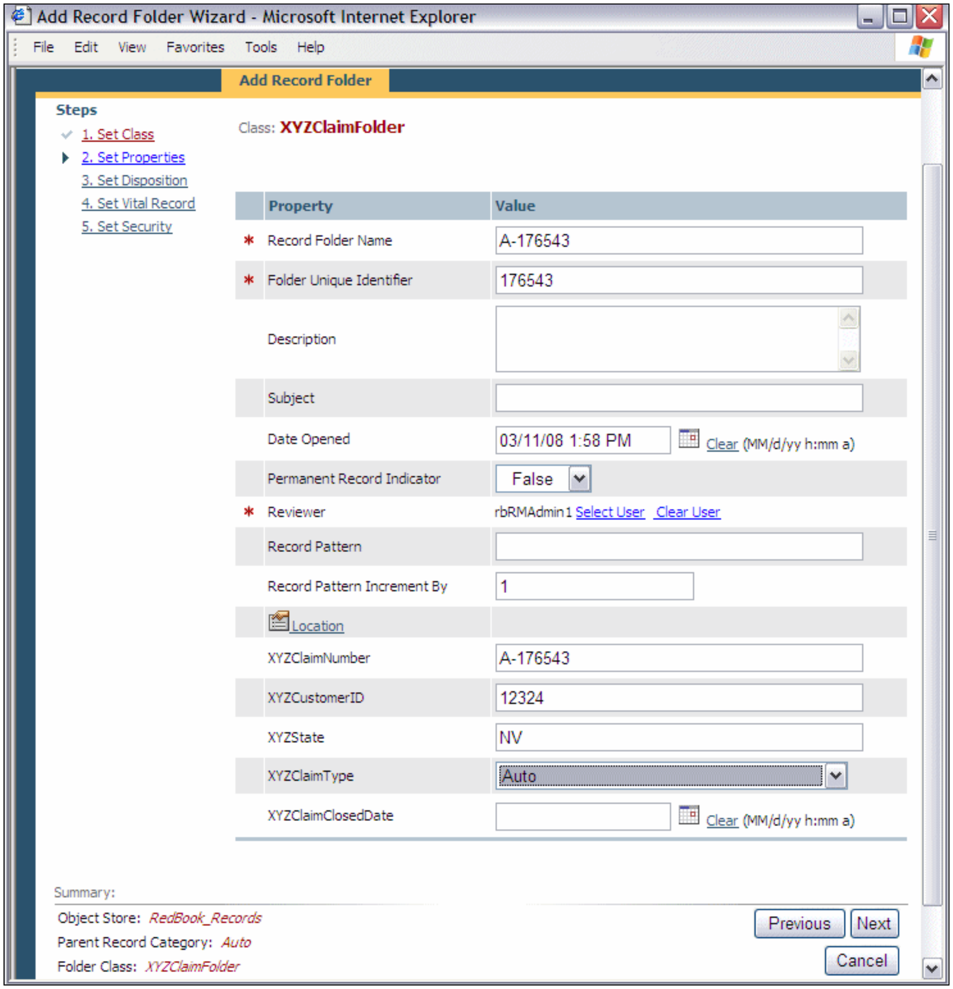
Summary:
Object Store: *RedBook_Records*
Parent Record Category: *Auto*
Folder Class: *Record Folder*

Figure 11-53 XYZ File Plan Add Record Folder: Set Class

5. Enter the metadata for the folder and click **Next**. For our case study, we enter:

   – Record Folder Name: A-176543
   – Folder Unique Identifier: 176543
   – Custom properties:
      - XYZClaimNumber: A-176543
      - XYZCustomerID: 12324
      - XYZState: NV
      - XYZClaimType: Auto

   Refer to Figure 11-54.


Add Record Folder Wizard - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Add Record Folder

Steps
1. Set Class
2. Set Properties
3. Set Disposition
4. Set Vital Record
5. Set Security

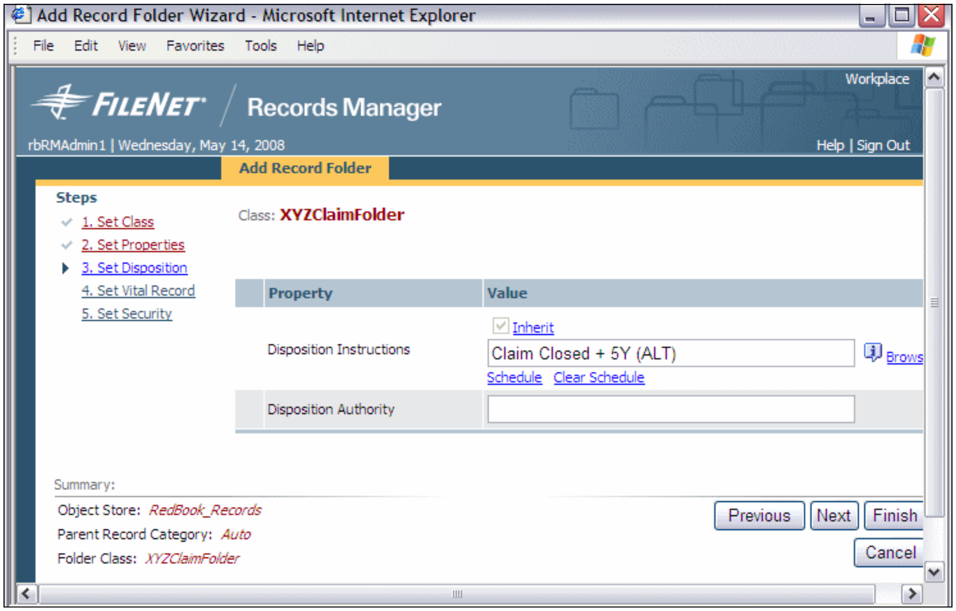
Class: **XYZClaimFolder**

	Property	Value
*	Record Folder Name	A-176543
*	Folder Unique Identifier	176543
	Description	
	Subject	
	Date Opened	03/11/08 1:58 PM Clear (MM/d/yy h:mm a)
	Permanent Record Indicator	False
*	Reviewer	rbRMAdmin1 Select User Clear User
	Record Pattern	
	Record Pattern Increment By	1
	Location	
	XYZClaimNumber	A-176543
	XYZCustomerID	12324
	XYZState	NV
	XYZClaimType	Auto
	XYZClaimClosedDate	Clear (MM/d/yy h:mm a)

Summary:
Object Store: *RedBook_Records*
Parent Record Category: *Auto*
Folder Class: *XYZClaimFolder*

Figure 11-54 XYZ File Plan Add Record Folder: Set Properties

6. If necessary, set or override the disposition schedule and click **Next**. For this case study, we want the Claim Folder to inherit its disposition schedule (refer to Figure 11-55). If the Vital Record and Security information do not need to be updated, click **Finish**. For the case study, these values do not need to be updated, but we show you these windows.



Class: XYZClaimFolder

Property	Value
Disposition Instructions	☑ Inherit Claim Closed + 5Y (ALT)
Disposition Authority	

Summary:

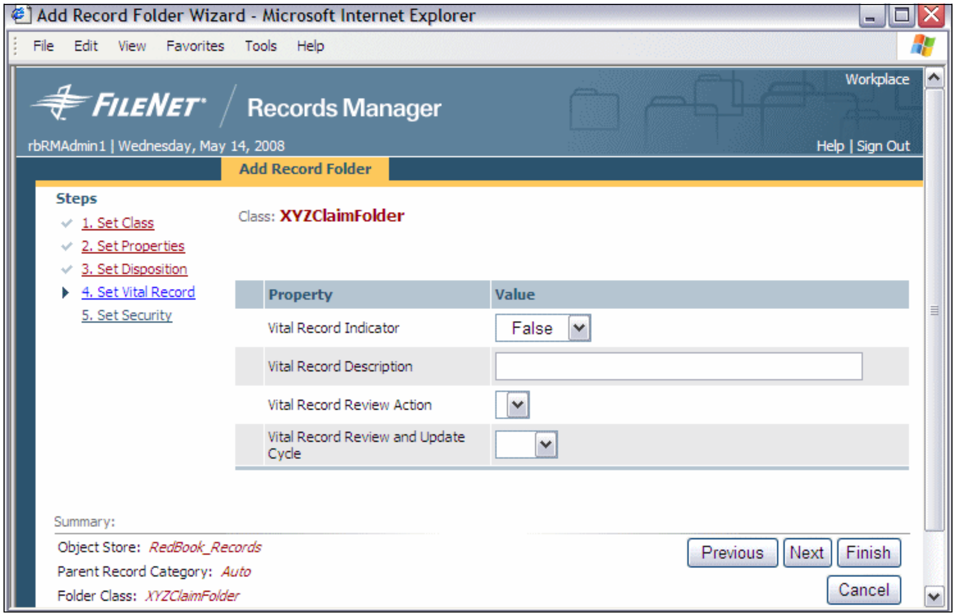
Object Store: *RedBook_Records*

Parent Record Category: *Auto*

Folder Class: *XYZClaimFolder*

Figure 11-55 XYZ File Plan Add Record Folder: Set Disposition

7. Set Vital Record information if necessary and click **Next**. Refer to Figure 11-56 on page 299.



Class: **XYZClaimFolder**

Property	Value
Vital Record Indicator	False
Vital Record Description	
Vital Record Review Action	
Vital Record Review and Update Cycle	

Summary:

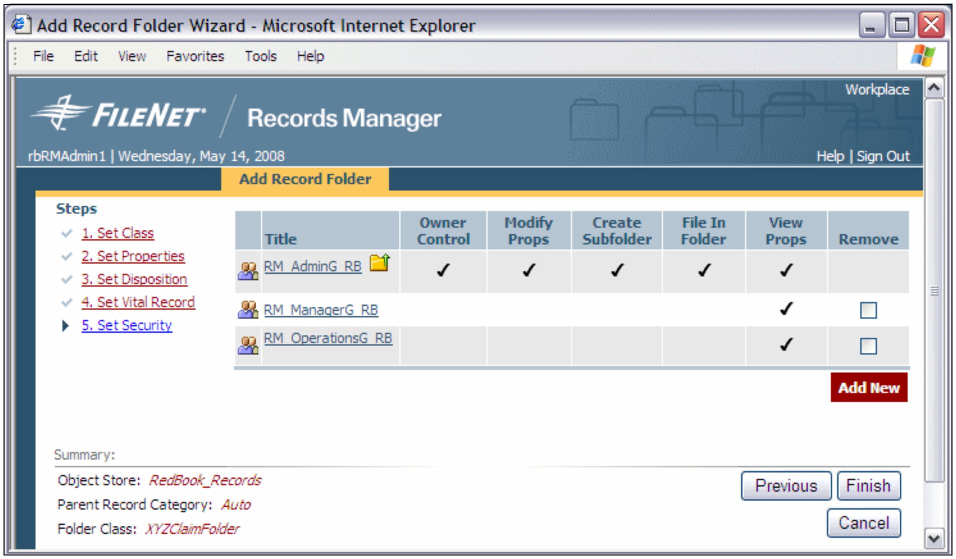
Object Store: *RedBook_Records*

Parent Record Category: *Auto*

Folder Class: *XYZClaimFolder*

Figure 11-56 XYZ File Plan Add Record Folder: Set Vital Record

8. Set Security information if necessary and click **Finish**. Refer to Figure 11-57.



Title	Owner Control	Modify Props	Create Subfolder	File In Folder	View Props	Remove
RM_AdminG_RB	✓	✓	✓	✓	✓	
RM_ManagerG_RB					✓	□
RM_OperationsG_RB					✓	□

Summary:

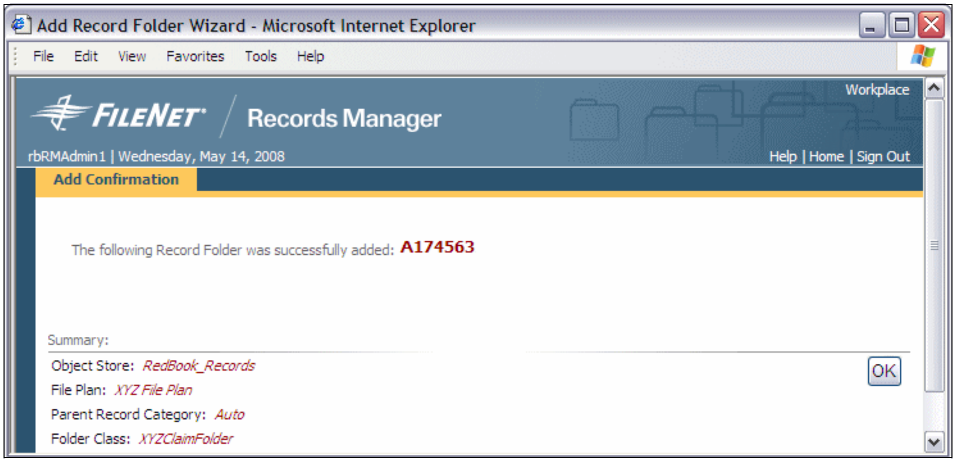
Object Store: *RedBook_Records*

Parent Record Category: *Auto*

Folder Class: *XYZClaimFolder*

Figure 11-57 XYZ File Plan Add Record Folder: Set Security

9. Click **OK** on the Add Confirmation window. Refer to Figure 11-58.


Add Record Folder Wizard - Microsoft Internet Explorer
File Edit View Favorites Tools Help
FILENET Records Manager
Workplace
Help | Home | Sign Out
Add Confirmation
OK

rbRMAdmin1 | Wednesday, May 14, 2008

The following Record Folder was successfully added: **A174563**

Summary:

Object Store: *RedBook_Records*

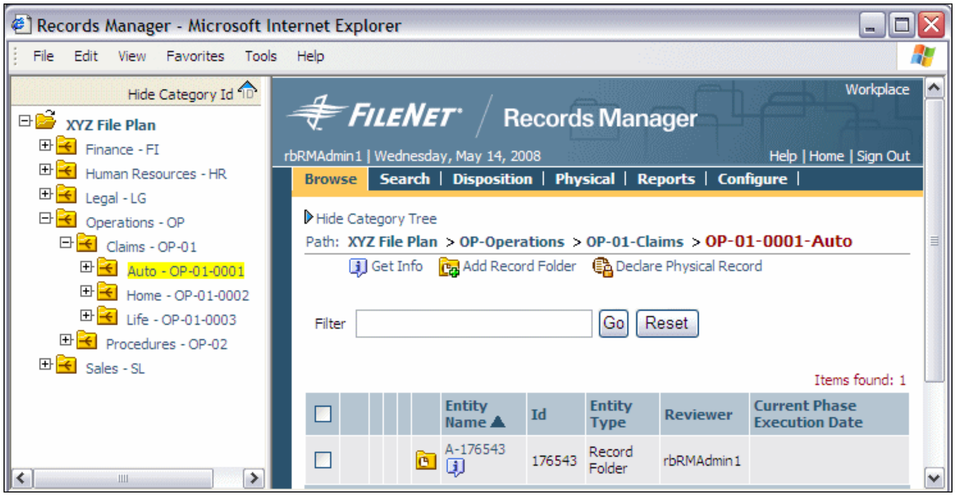
File Plan: *XYZ File Plan*

Parent Record Category: *Auto*

Folder Class: *XYZClaimFolder*

Figure 11-58 XYZ File Plan Add Record Folder was successfully completed

When you are returned to the file plan window, the new folder is visible. Refer to Figure 11-59.


Records Manager - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Hide Category Id
FILENET Records Manager
Workplace
Help | Home | Sign Out
Browse | Search | Disposition | Physical | Reports | Configure |
Hide Category Tree
Get Info
Add Record Folder
Declare Physical Record
Go
Reset

- XYZ File Plan
  - Finance - FI
  - Human Resources - HR
  - Legal - LG
  - Operations - OP
    - Claims - OP-01
      - Auto - OP-01-0001
      - Home - OP-01-0002
      - Life - OP-01-0003
    - Procedures - OP-02
  - Sales - SL

rbRMAdmin1 | Wednesday, May 14, 2008

Path: **XYZ File Plan > OP-Operations > OP-01-Claims > OP-01-0001-Auto**

Filter

Items found: 1

□				Entity Name ▲	Id	Entity Type	Reviewer	Current Phase Execution Date
□				A-176543	176543	Record Folder	rbRMAdmin1	

Figure 11-59 XYZ File Plan New folder added

12

Records declaration case study

In Chapter 11, “File plan creation case study” on page 255, we showed you how to build the file plan for our case study. In this chapter, we describe the steps that are required to build entry templates for records declaration.

We describe the following topics in this chapter for building entry templates for declaration:

- Creating a Declare as Record Template (in the ROS)
- Creating a Document Entry Template (in the ROS)
- Declare a record using the entry template
- Building templates for other contract types

12.1 Building entry templates for declaration

Our case study is based on the fictitious Fictional Insurance Company X that we introduced in 3.3.2, “Example file plan” on page 65. The company’s file plan is shown in Figure 3-3 on page 66. The partial file plan that shows categories that are related to the case study for this book is illustrated in Figure 12-1.

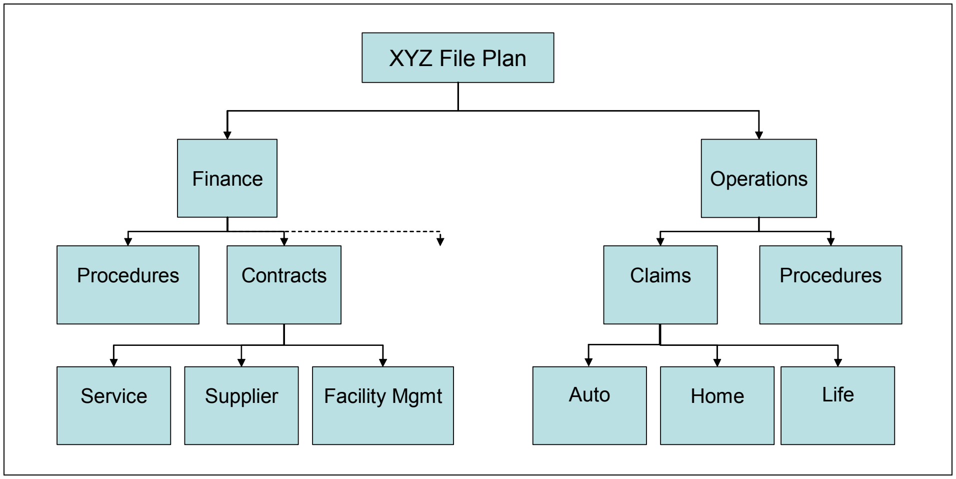


Figure 12-1 Partial file plan showing categories related to the case study for this book

The XYZ File Plan includes a separate category for each of the three contract types:

- Facility Mgmt: Facility Management Contracts
- Service: Service Contracts
- Supplier: Supplier Contracts

If we want to use entry templates as a mechanism for records declaration, and if we want to automate the declaration process, one approach is to have three separate sets of declare/entry templates, one pair for each of the three contract types. This approach allows a user who wants to add a new contract document to simply select the correct entry template based on which of the three contract types the user wants to add.

Before you can declare records with entry templates, you have to create the appropriate templates. In this section, we show you how to perform the following tasks:

- Create a *Declare As Record Template* in the Records-enabled content Object Store (ROS) for Facility Management Contract documents.
- Create a document entry template for Facility Management Contract Docs that uses the *Declare As Record Template*.
- Use the document entry template to add a new document in Workplace and have that document automatically be declared as a record into the correct file plan location to verify that the template is working correctly.

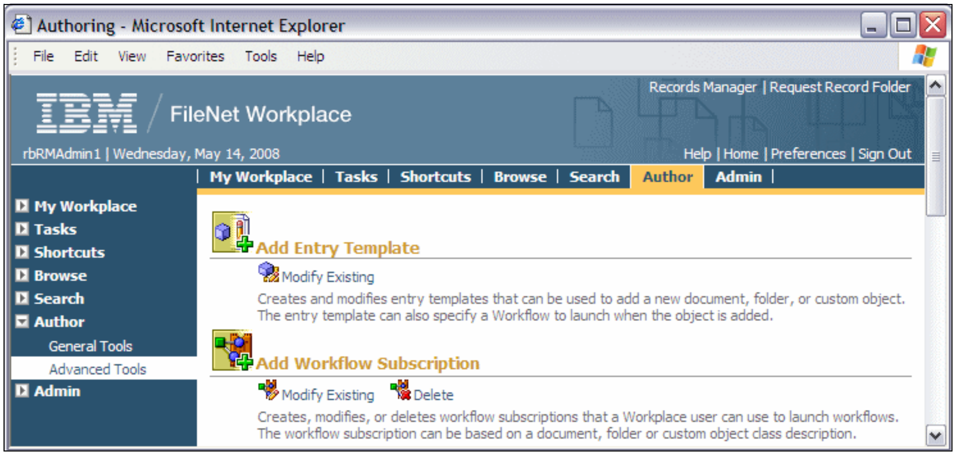
12.2 Creating a Declare as Record Template (in the ROS)

The steps to create a Declare as Record Template are:

1. Log on to Workplace with advanced author permissions.
2. Launch the Declare Records Template Designer.
3. Configure the default selections for record class, file plan location, and record properties.
4. Save the declare as record template so that it can be used by the document entry template.

To create a Declare as Record Template, follow these instructions:

1. Launch the Declare Records Entry Template Designer:
    a. Log on to Workplace with advanced author permissions.
    b. Select **Author** → **Advanced Tools**. Figure 12-2 on page 304 displays.


Authoring - Microsoft Internet Explorer
File Edit View Favorites Tools Help
IBM FileNet Workplace
Records Manager | Request Record Folder
rbRMAdmin1 | Wednesday, May 14, 2008
Help | Home | Preferences | Sign Out
My Workplace | Tasks | Shortcuts | Browse | Search | Author | Admin
My Workplace
Tasks
Shortcuts
Browse
Search
Author
General Tools
Advanced Tools
Admin
Add Entry Template
Modify Existing
Add Workflow Subscription
Modify Existing
Delete

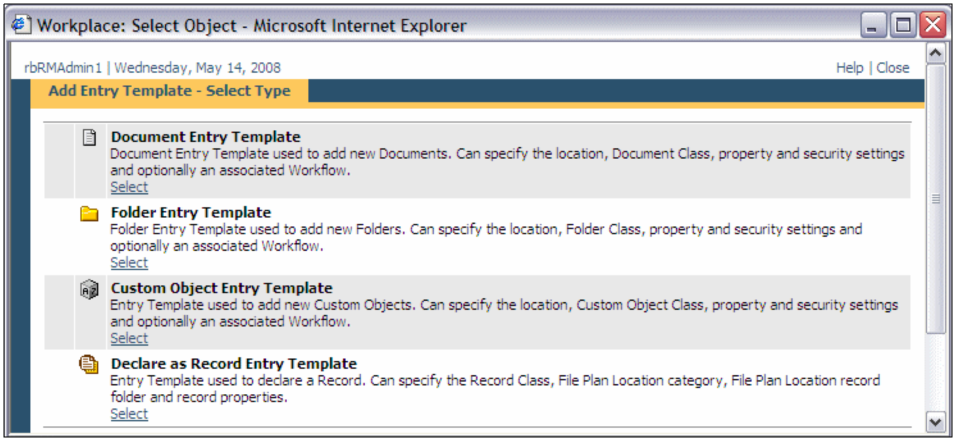
Creates and modifies entry templates that can be used to add a new document, folder, or custom object. The entry template can also specify a Workflow to launch when the object is added.

Creates, modifies, or deletes workflow subscriptions that a Workplace user can use to launch workflows. The workflow subscription can be based on a document, folder or custom object class description.

Figure 12-2 Workplace → Author → Advanced Tools: Select Add Entry Template

c. Select **Add Entry Template** from Figure 12-2.

d. Select **Declare as Record Entry Template** from Figure 12-3, which launches Declare Records Template Designer.


Workplace: Select Object - Microsoft Internet Explorer
rbRMAdmin1 | Wednesday, May 14, 2008
Help | Close
Add Entry Template - Select Type

Document Entry Template
Document Entry Template used to add new Documents. Can specify the location, Document Class, property and security settings and optionally an associated Workflow.
Select

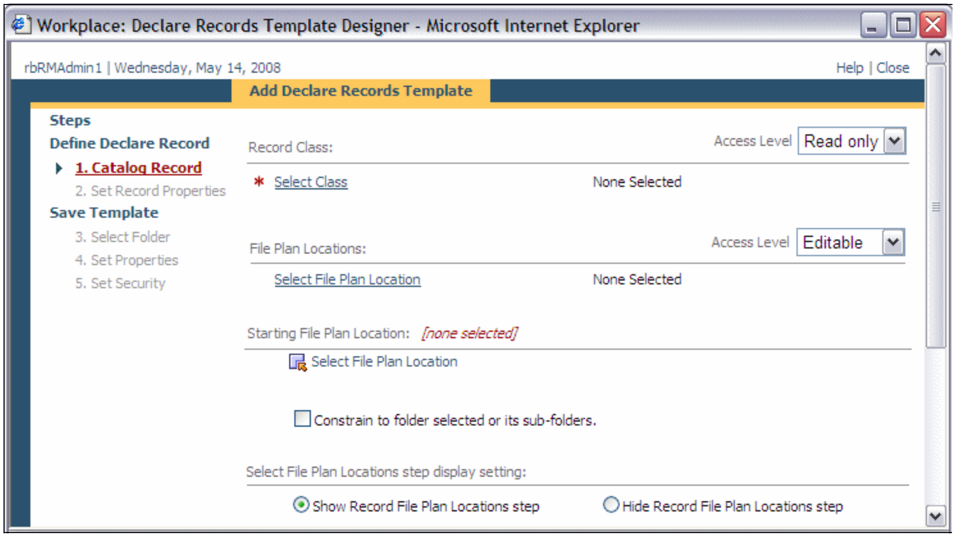
Folder Entry Template
Folder Entry Template used to add new Folders. Can specify the location, Folder Class, property and security settings and optionally an associated Workflow.
Select

Custom Object Entry Template
Entry Template used to add new Custom Objects. Can specify the location, Custom Object Class, property and security settings and optionally an associated Workflow.
Select

Declare as Record Entry Template
Entry Template used to declare a Record. Can specify the Record Class, File Plan Location category, File Plan Location record folder and record properties.
Select

Figure 12-3 Workplace → Entry Template: Select Declare as Record Entry Template

Declare Records Template Designer launches. Refer to Figure 12-4 on page 305.



Workplace: Declare Records Template Designer - Microsoft Internet Explorer

rbRMAdmin1 | Wednesday, May 14, 2008 Help | Close

Add Declare Records Template

Steps

Define Declare Record

- 1. Catalog Record
- 2. Set Record Properties

Save Template

- 3. Select Folder
- 4. Set Properties
- 5. Set Security

Record Class: Access Level Read only

* Select Class None Selected

File Plan Locations: Access Level Editable

Select File Plan Location None Selected

Starting File Plan Location: *[none selected]*

Select File Plan Location

□ Constrain to folder selected or its sub-folders.

Select File Plan Locations step display setting:

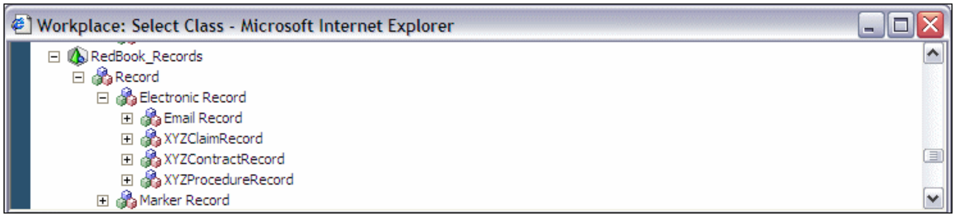
● Show Record File Plan Locations step ○ Hide Record File Plan Locations step

Figure 12-4 Declare Records Template: Catalog Record

2. Select the record class into which documents are to be declared:

   a. Click **Select Class**.

   b. Navigate to the class into which you want to declare documents. For our case study, we expand the FPOS (RedBook_Records) and select **Record** → **Electronic Record** → **XYZContractRecord** class. Refer to Figure 12-5.

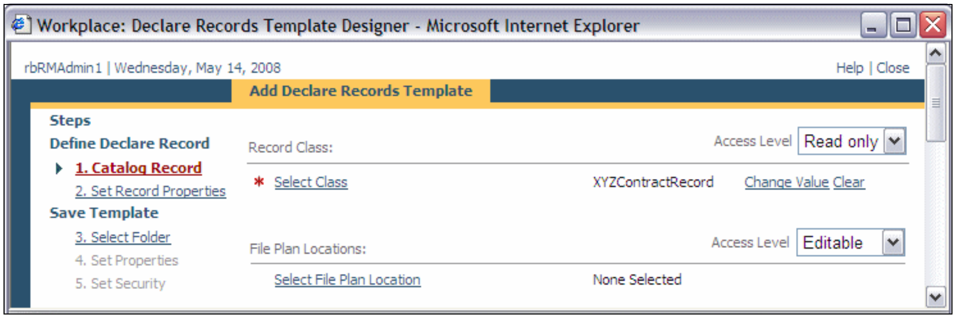


Workplace: Select Class - Microsoft Internet Explorer

```
⊟ RedBook_Records
  ⊟ Record
    ⊟ Electronic Record
      ⊞ Email Record
      ⊞ XYZClaimRecord
      ⊞ XYZContractRecord
      ⊞ XYZProcedureRecord
    ⊞ Marker Record
```

Figure 12-5 Declare Records Template: Catalog Record: Browse record classes

Figure 12-6 on page 306 shows the class being selected.



Workplace: Declare Records Template Designer - Microsoft Internet Explorer

rbRMAdmin1 | Wednesday, May 14, 2008 Help | Close

Add Declare Records Template

Steps
Define Declare Record
1. Catalog Record
2. Set Record Properties
Save Template
3. Select Folder
4. Set Properties
5. Set Security

Record Class: Access Level Read only

* Select Class XYZContractRecord Change Value Clear

File Plan Locations: Access Level Editable

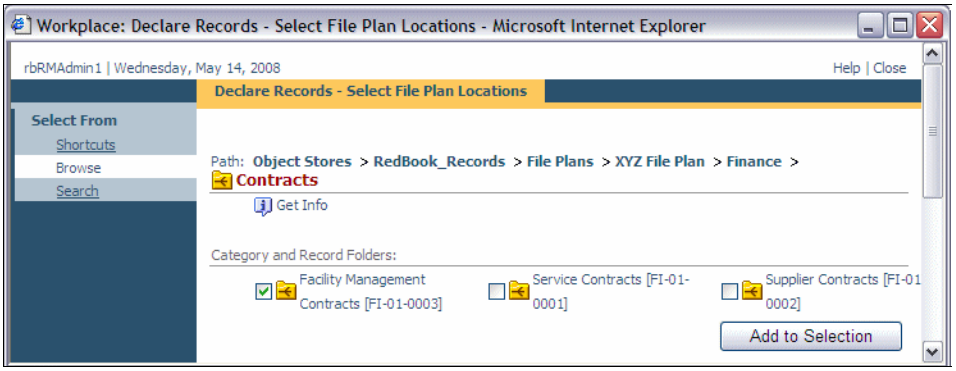
Select File Plan Location None Selected

Figure 12-6 Declare Records Template: Catalog Record: Record class selected

3. Select the file plan location into which the documents are to be declared:
   a. Click **Select File Plan Location**.
   b. Navigate to the location where you want to declare documents into.

      For our case study, we select the FPOS (RedBook_Records) and select **Finance** → **Contracts**. Refer to Figure 12-7.
   c. Select the file plan category and click **Add to Selection**.

      For our case study, we select **Facilities Management Contracts**.



Workplace: Declare Records - Select File Plan Locations - Microsoft Internet Explorer

rbRMAdmin1 | Wednesday, May 14, 2008 Help | Close

Declare Records - Select File Plan Locations

Select From
Shortcuts
Browse
Search

Path: Object Stores > RedBook_Records > File Plans > XYZ File Plan > Finance >
Contracts
Get Info

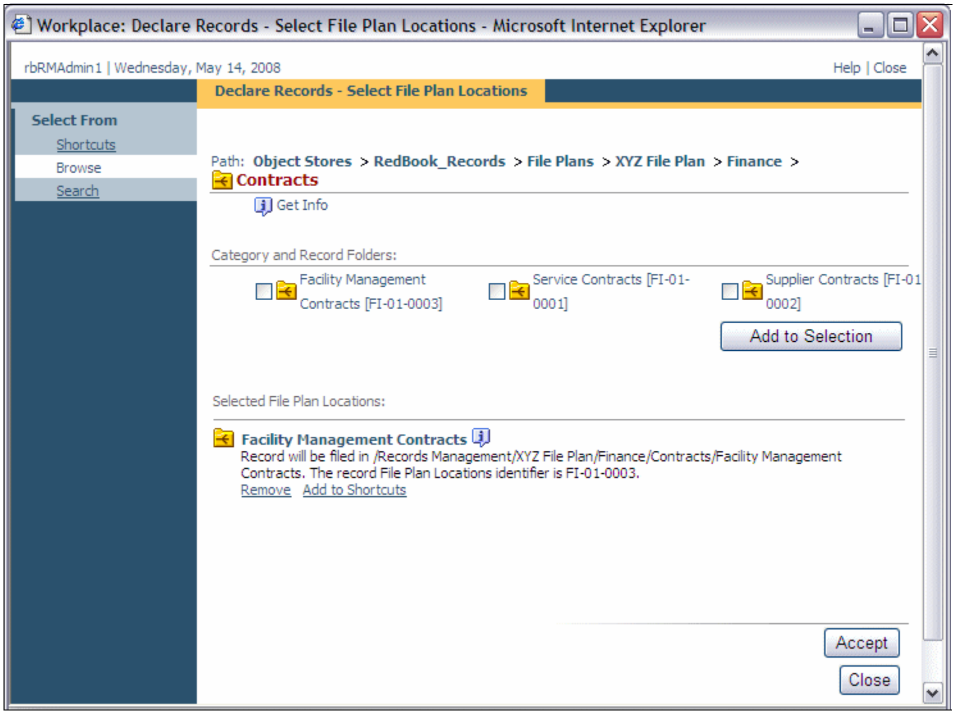
Category and Record Folders:

☑ Facility Management Contracts [FI-01-0003]
□ Service Contracts [FI-01-0001]
□ Supplier Contracts [FI-01 0002]

Add to Selection

Figure 12-7 Declare Records Template: Catalog Record: Select File Plan Locations

The Facility Management Contract category is added to the list of selected file plan locations. Refer to Figure 12-8 on page 307.



Workplace: Declare Records - Select File Plan Locations - Microsoft Internet Explorer

rbRMAdmin1 | Wednesday, May 14, 2008

Help | Close

Declare Records - Select File Plan Locations

Select From

Shortcuts

Browse

Search

Path: Object Stores > RedBook_Records > File Plans > XYZ File Plan > Finance >
Contracts

Get Info

Category and Record Folders:

□ Facility Management Contracts [FI-01-0003]	□ Service Contracts [FI-01-0001]	□ Supplier Contracts [FI-01 0002]

Add to Selection

Selected File Plan Locations:

Facility Management Contracts
Record will be filed in /Records Management/XYZ File Plan/Finance/Contracts/Facility Management Contracts. The record File Plan Locations identifier is FI-01-0003.
Remove Add to Shortcuts

Accept

Close

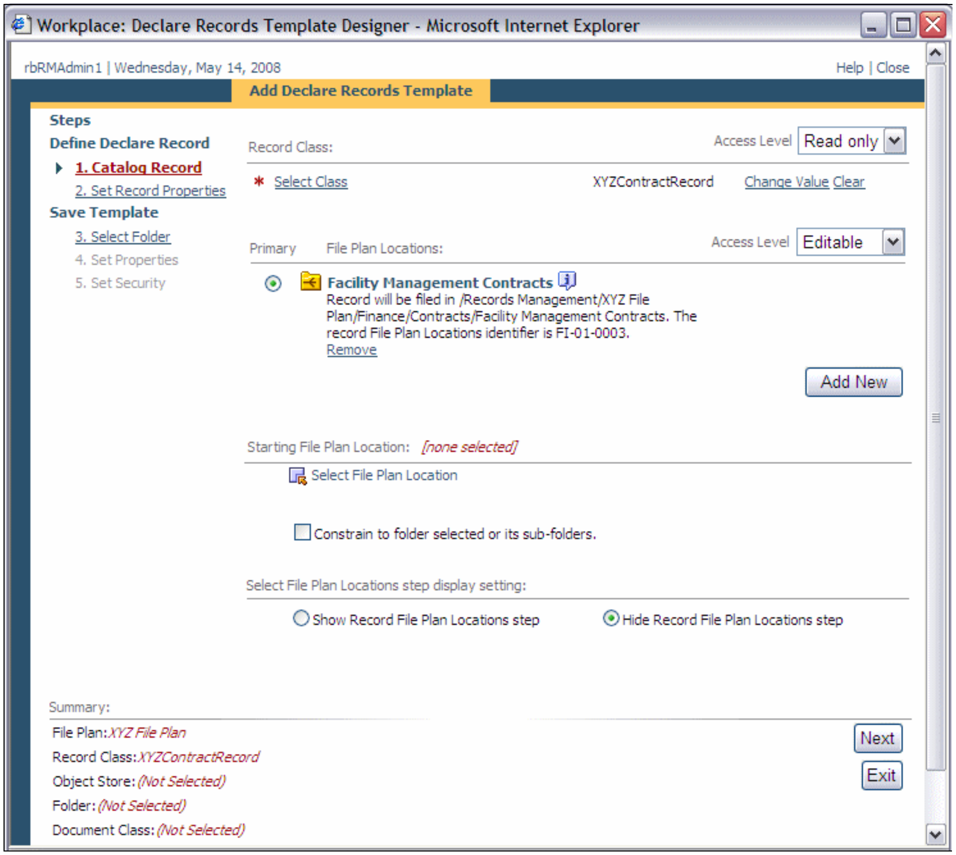
Figure 12-8 Declare Records Template: Catalog Record: File plan location selected

d. Click **Accept**.

4. Select **Hide Record File Plan Locations step** (refer to Figure 12-9 on page 308) and click **Next**.

   When hiding the file plan location step, users will not see the file plan location when they declare documents as records. We used this option, because our case study calls for having a separate entry template for each type of contract record. We know exactly where in the file plan location to file the records. Setting this hide option eliminates users having to select the file plan category, because the correct category is preselected.

   The best practice is always to eliminate as much user decision making as possible when declaring records into the system, which results in the records declaration process being more efficient and less error-prone.


Workplace: Declare Records Template Designer - Microsoft Internet Explorer

rbRMAdmin1 | Wednesday, May 14, 2008

Help | Close

Add Declare Records Template

Steps

Define Declare Record

- 1. Catalog Record
- 2. Set Record Properties

Save Template

- 3. Select Folder
- 4. Set Properties
- 5. Set Security

Record Class: Access Level: Read only

* Select Class XYZContractRecord Change Value Clear

Primary File Plan Locations: Access Level: Editable

● Facility Management Contracts
Record will be filed in /Records Management/XYZ File Plan/Finance/Contracts/Facility Management Contracts. The record File Plan Locations identifier is FI-01-0003.
Remove

Add New

Starting File Plan Location: *[none selected]*

Select File Plan Location

□ Constrain to folder selected or its sub-folders.

Select File Plan Locations step display setting:

○ Show Record File Plan Locations step ● Hide Record File Plan Locations step

Summary:

File Plan: *XYZ File Plan*

Record Class: *XYZContractRecord*

Object Store: *(Not Selected)*

Folder: *(Not Selected)*

Document Class: *(Not Selected)*

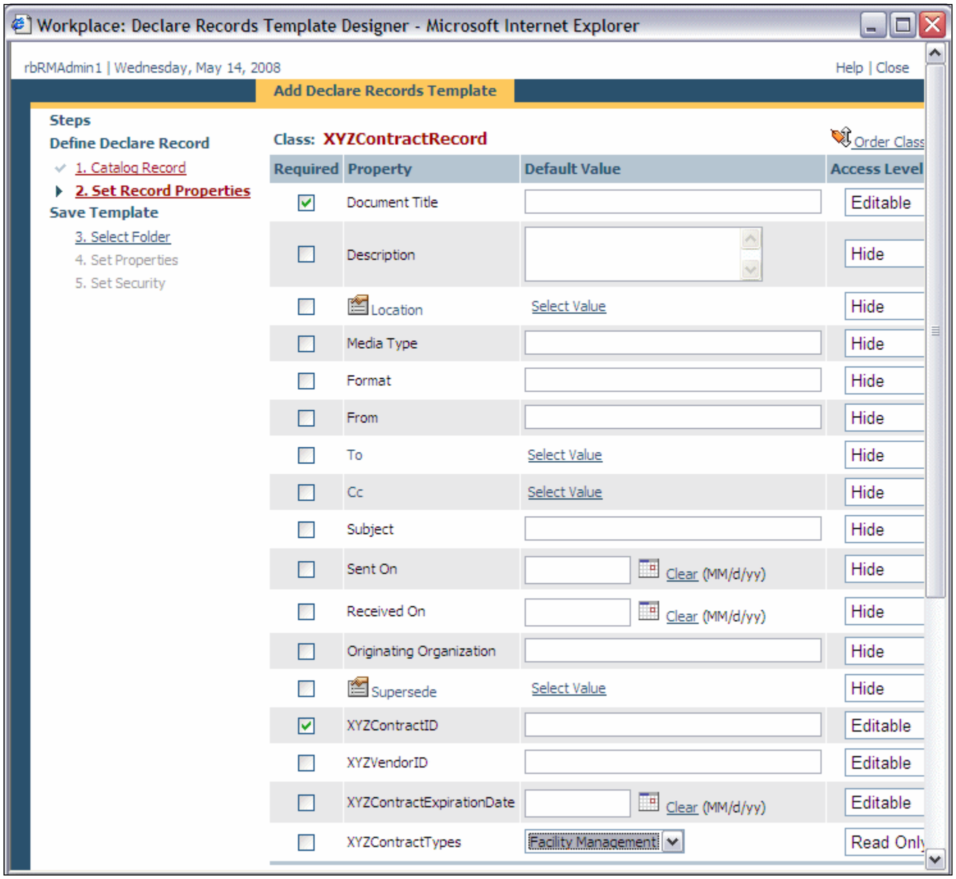
Next

Exit

Figure 12-9 Declare Records Template: Catalog Record: Hide Record File Plan Locations step

5. Set record properties and click **Next**.

   This best practice is to hide the fields that users do not need to edit and see, set mandatory fields as required, and specify default values if applicable.



Workplace: Declare Records Template Designer - Microsoft Internet Explorer

rbRMAdmin1 | Wednesday, May 14, 2008

Help | Close

Add Declare Records Template

Steps

Define Declare Record

✓ 1. Catalog Record

2. Set Record Properties

Save Template

3. Select Folder

4. Set Properties

5. Set Security

Class: XYZContractRecord

Order Class

Required	Property	Default Value	Access Level
☑	Document Title		Editable
□	Description		Hide
□	Location	Select Value	Hide
□	Media Type		Hide
□	Format		Hide
□	From		Hide
□	To	Select Value	Hide
□	Cc	Select Value	Hide
□	Subject		Hide
□	Sent On	Clear (MM/d/yy)	Hide
□	Received On	Clear (MM/d/yy)	Hide
□	Originating Organization		Hide
□	Supersede	Select Value	Hide
☑	XYZContractID		Editable
□	XYZVendorID		Editable
□	XYZContractExpirationDate	Clear (MM/d/yy)	Editable
□	XYZContractTypes	Facility Management	Read Only

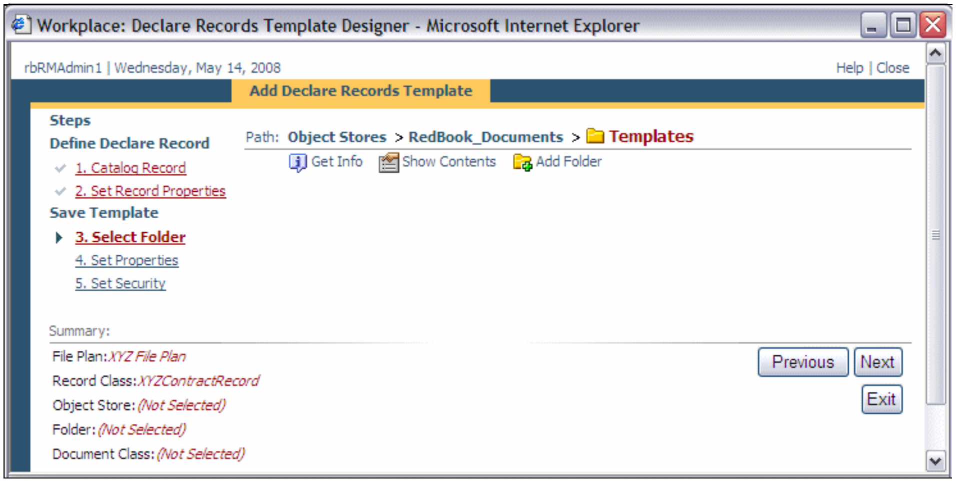
Figure 12-10 Declare Records Template: Set Record Properties

For our case study, we set these properties (refer to Figure 12-10):

- Document Title: Set to be required and editable.
- XYZContractID: Set to be required and editable.
- XYZVendorID: Set to be editable.
- XYZContractExpirationDate: Set to be editable.
- XYZContractType: Select **Facility Management.** Set to Read Only.
- Select **Show Set Record Properties step** (not shown in Figure 12-10).

6. Select a folder in the ROS where you want to store the template, and click **Next**.

   For our case study, we navigate to **Redbook_Documents** and select the **Templates** folder (refer to Figure 12-11 on page 310).


Workplace: Declare Records Template Designer - Microsoft Internet Explorer
Help | Close
Get Info
Show Contents
Add Folder
Previous
Next
Exit

rbRMAdmin1 | Wednesday, May 14, 2008

Add Declare Records Template

Steps
Define Declare Record
✓ 1. Catalog Record
✓ 2. Set Record Properties
Save Template
▸ 3. Select Folder
4. Set Properties
5. Set Security

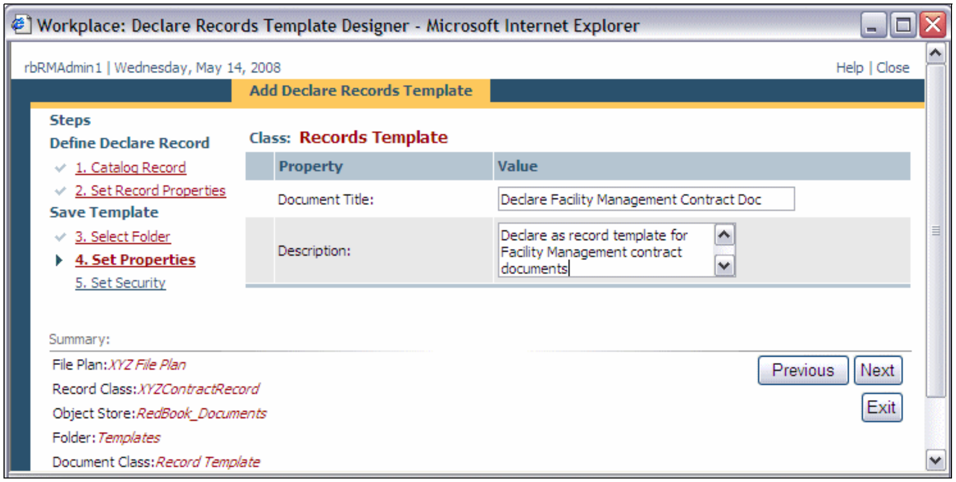
Path: Object Stores > RedBook_Documents > Templates

Summary:
File Plan: *XYZ File Plan*
Record Class: *XYZContractRecord*
Object Store: *(Not Selected)*
Folder: *(Not Selected)*
Document Class: *(Not Selected)*

Figure 12-11 Declare Records Template: Select Folder

7. Set the template properties by entering the entry template name (Document Title) and description, and click **Next**.

   For our case study, we name the template: Declare Facility Management Contract Doc. Refer to Figure 12-12.


Workplace: Declare Records Template Designer - Microsoft Internet Explorer
Help | Close
Previous
Next
Exit

rbRMAdmin1 | Wednesday, May 14, 2008

Add Declare Records Template

Steps
Define Declare Record
✓ 1. Catalog Record
✓ 2. Set Record Properties
Save Template
✓ 3. Select Folder
▸ 4. Set Properties
5. Set Security

Class: Records Template

Property	Value
Document Title:	Declare Facility Management Contract Doc
Description:	Declare as record template for Facility Management contract documents

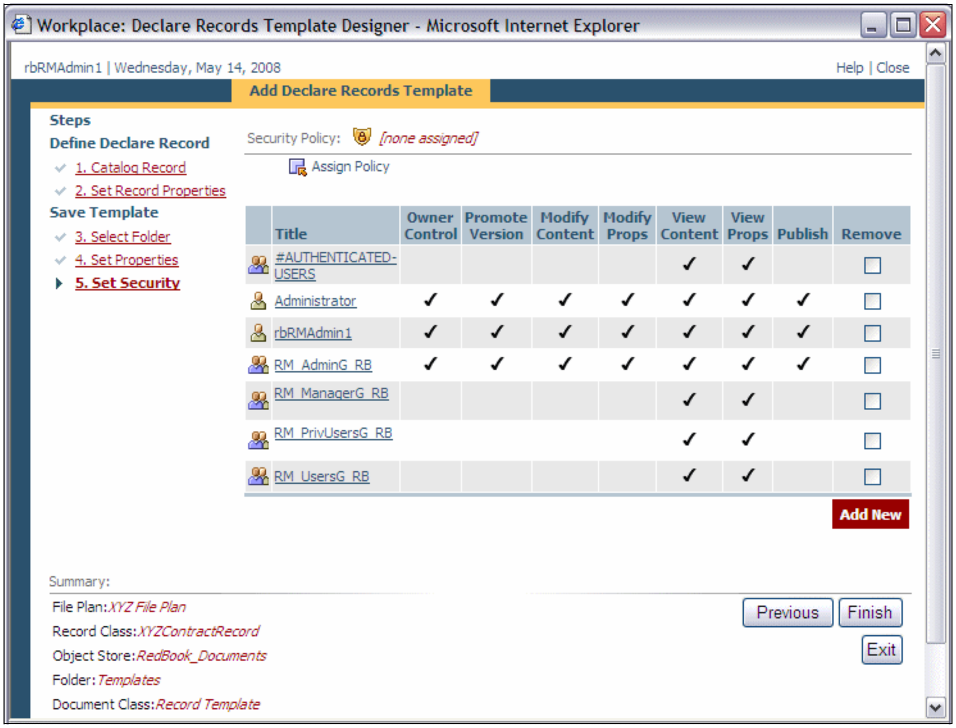
Summary:
File Plan: *XYZ File Plan*
Record Class: *XYZContractRecord*
Object Store: *RedBook_Documents*
Folder: *Templates*
Document Class: *Record Template*

Figure 12-12 Declare Records Template: Set Properties

8. Set the security for this entry template.

   Security determines which users have permission to use this template. The choices that you make in this step are determined by your overall security schema and requirements.

For our case study, we use the default security choices. Refer to Figure 12-13.



Workplace: Declare Records Template Designer - Microsoft Internet Explorer

rbRMAdmin1 | Wednesday, May 14, 2008 Help | Close

Add Declare Records Template

Steps

Define Declare Record

✓ 1. Catalog Record

✓ 2. Set Record Properties

Save Template

✓ 3. Select Folder

✓ 4. Set Properties

▶ 5. Set Security

Security Policy: [none assigned]

Assign Policy

	Title	Owner Control	Promote Version	Modify Content	Modify Props	View Content	View Props	Publish	Remove
	#AUTHENTICATED-USERS					✓	✓		□
	Administrator	✓	✓	✓	✓	✓	✓	✓	□
	rbRMAdmin1	✓	✓	✓	✓	✓	✓	✓	□
	RM_AdminG_RB	✓	✓	✓	✓	✓	✓	✓	□
	RM_ManagerG_RB					✓	✓		□
	RM_PrivUsersG_RB					✓	✓		□
	RM_UsersG_RB					✓	✓		□

Add New

Summary:

File Plan: *XYZ File Plan*

Record Class: *XYZContractRecord*

Object Store: *RedBook_Documents*

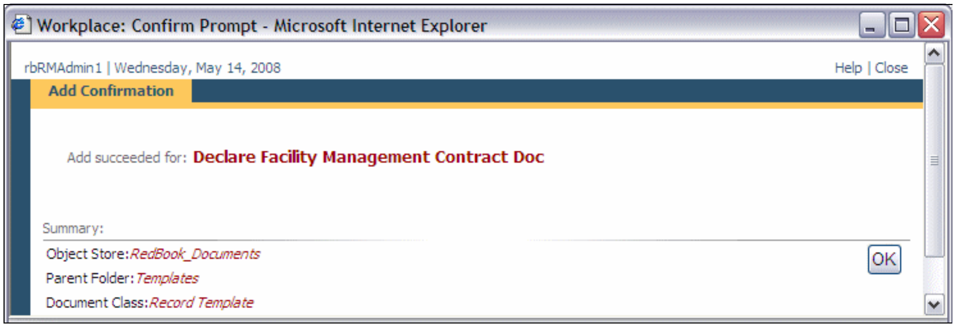
Folder: *Templates*

Document Class: *Record Template*

Figure 12-13 Declare Records Template: Set Security

9. Click **Finish**, and then, click **OK** to finish the entry template setup.

   Figure 12-14 shows the confirmation message telling us that the template has been successfully added to the repository.



Workplace: Confirm Prompt - Microsoft Internet Explorer

rbRMAdmin1 | Wednesday, May 14, 2008 Help | Close

Add Confirmation

Add succeeded for: **Declare Facility Management Contract Doc**

Summary:

Object Store: *RedBook_Documents*

Parent Folder: *Templates*

Document Class: *Record Template*

Figure 12-14 Declare Records Template: Confirmation message

12.3 Creating a Document Entry Template (in the ROS)

The steps to create a Document Entry Template are:

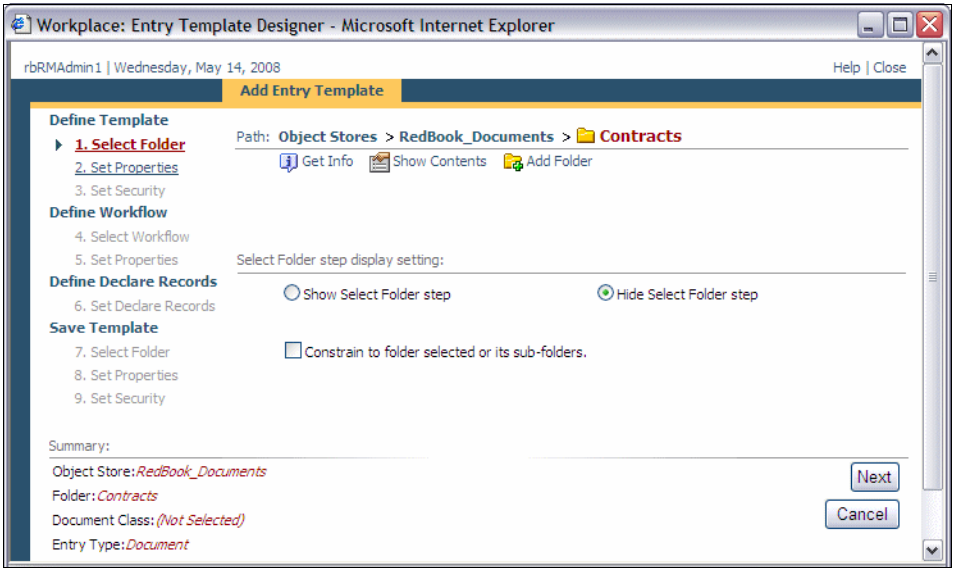
1. Log on to Workplace with advanced author permissions.
2. Launch the Document Entry Template Designer.
3. Configure the default selections for folders, document class, and properties.
4. Link the Document Entry Template to the Declare As Record Template so the two entry templates work together for records declaration.
5. Save the template so that it can be used from Workplace.

To create a Document Entry Template, following these instructions:

1. Launch the Document Entry Template Designer:
   a. Log on to Workplace with advanced author permissions.
   b. Select **Author** → **Advanced Tools** → **Add Entry Template** → **Document Entry Template**, which launches the Document Entry Template Designer.
2. Select the ROS folder options as shown in Figure 12-15 on page 313:
   a. Navigate to the folder in the ROS where you want the document to be added.

      For our case study, we select **Redbook_Documents** → **Contracts**. Note, we want all contract documents to go into a folder called Contracts.

   b. Choose **Hide Select Folder step**, because we do not want the user to change this folder selection.
   c. Click **Next**.


Workplace: Entry Template Designer - Microsoft Internet Explorer

rbRMAdmin1 | Wednesday, May 14, 2008 — Help | Close

Add Entry Template

Define Template
- 1. Select Folder
- 2. Set Properties
- 3. Set Security

Define Workflow
- 4. Select Workflow
- 5. Set Properties

Define Declare Records
- 6. Set Declare Records

Save Template
- 7. Select Folder
- 8. Set Properties
- 9. Set Security

Path: Object Stores > RedBook_Documents > Contracts

Get Info | Show Contents | Add Folder

Select Folder step display setting:

○ Show Select Folder step ● Hide Select Folder step

□ Constrain to folder selected or its sub-folders.

Summary:

Object Store: *RedBook_Documents*

Folder: *Contracts*

Document Class: *(Not Selected)*

Entry Type: *Document*

Figure 12-15 Entry Template: Select Folder

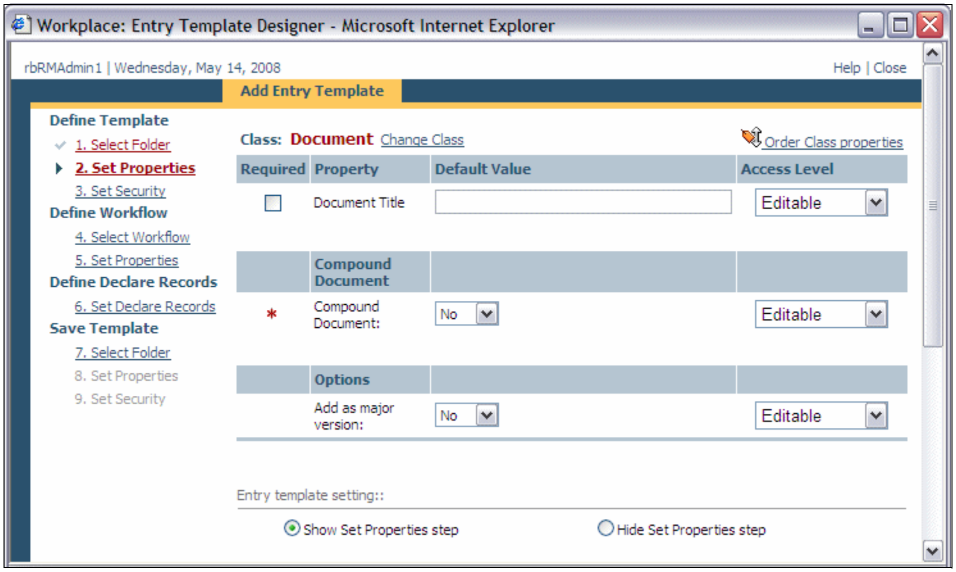
3. Set the entry template properties.

   The best practice is to hide the fields that users do not need to edit and see, set mandatory fields as required, and specify default values if applicable.

   Complete the following setup:

   a. Before setting the entry template properties, first change the class to the correct custom class if applicable. By default, our case study has the wrong class. Refer to Figure 12-16 on page 314:

      i. Select **Change Class**.



Workplace: Entry Template Designer - Microsoft Internet Explorer

rbRMAdmin1 | Wednesday, May 14, 2008 Help | Close

Add Entry Template

Define Template
- ✓ 1. Select Folder
- ▸ 2. Set Properties
- 3. Set Security

Define Workflow
- 4. Select Workflow
- 5. Set Properties

Define Declare Records
- 6. Set Declare Records

Save Template
- 7. Select Folder
- 8. Set Properties
- 9. Set Security

Class: Document Change Class Order Class properties

Required	Property	Default Value	Access Level
□	Document Title		Editable
	Compound Document		
*	Compound Document:	No	Editable
	Options		
	Add as major version:	No	Editable

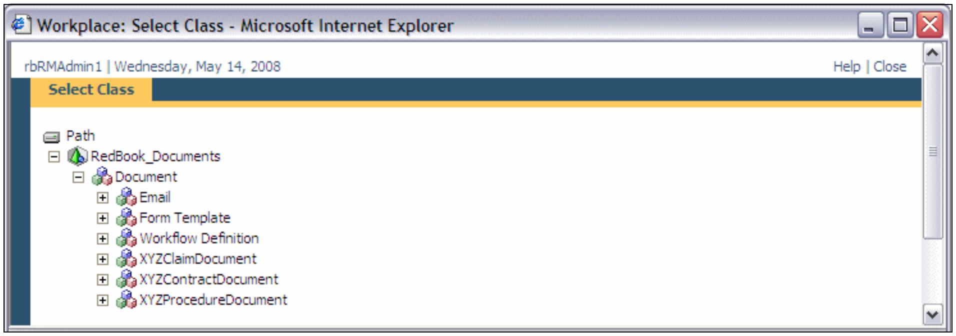
Entry template setting::

● Show Set Properties step ○ Hide Set Properties step

Figure 12-16 Entry Template: Set Properties: Change Class

ii. Select the correct class for the entry template.

For our case study, we select **XYZContractDocument** class from a selection of classes (refer to Figure 12-17).



Workplace: Select Class - Microsoft Internet Explorer

rbRMAdmin1 | Wednesday, May 14, 2008 Help | Close

Select Class

```
Path
⊟ RedBook_Documents
   ⊟ Document
      ⊞ Email
      ⊞ Form Template
      ⊞ Workflow Definition
      ⊞ XYZClaimDocument
      ⊞ XYZContractDocument
      ⊞ XYZProcedureDocument
```

Figure 12-17 Entry Template: Select Properties: Select a new class

b. Set the properties for the entry template.

For our case study, we set them as (refer to Figure 12-18 on page 315):

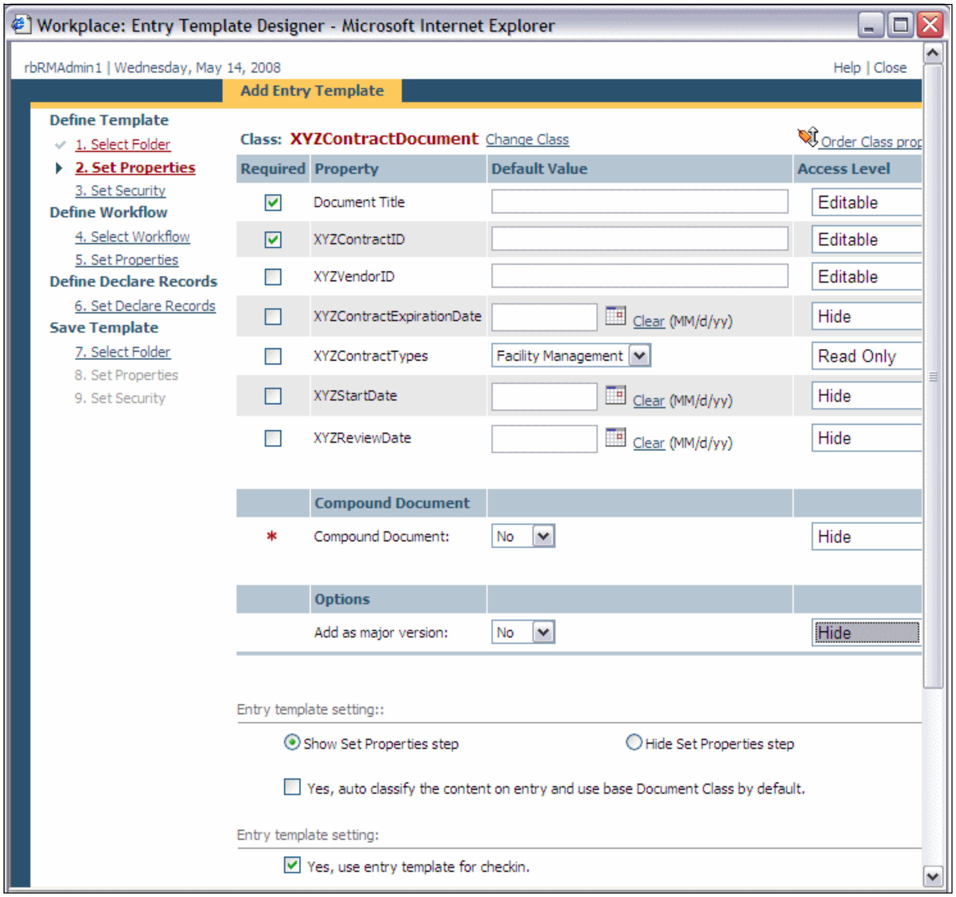
- Document Title: Set to be required and editable.
- XYZContractID: Set to be required and editable.
- XYZVendorID: Set to be editable.

- XYZContractType: Select **Facility Management.** Set to read only.

  Preselect XYZContractType with this value, because this entry template will only be used for Facility Management contracts.

- All other fields: Set to Hide.

c. Click **Next**.



Workplace: Entry Template Designer - Microsoft Internet Explorer

rbRMAdmin1 | Wednesday, May 14, 2008

Help | Close

Add Entry Template

Define Template
- ✓ 1. Select Folder
- 2. Set Properties
- 3. Set Security

Define Workflow
- 4. Select Workflow
- 5. Set Properties

Define Declare Records
- 6. Set Declare Records

Save Template
- 7. Select Folder
- 8. Set Properties
- 9. Set Security

Class: XYZContractDocument Change Class

Order Class prop

Required	Property	Default Value	Access Level
☑	Document Title		Editable
☑	XYZContractID		Editable
□	XYZVendorID		Editable
□	XYZContractExpirationDate	Clear (MM/d/yy)	Hide
□	XYZContractTypes	Facility Management	Read Only
□	XYZStartDate	Clear (MM/d/yy)	Hide
□	XYZReviewDate	Clear (MM/d/yy)	Hide
	Compound Document		
*	Compound Document:	No	Hide
	Options		
	Add as major version:	No	Hide

Entry template setting::

● Show Set Properties step ○ Hide Set Properties step

□ Yes, auto classify the content on entry and use base Document Class by default.

Entry template setting:

☑ Yes, use entry template for checkin.

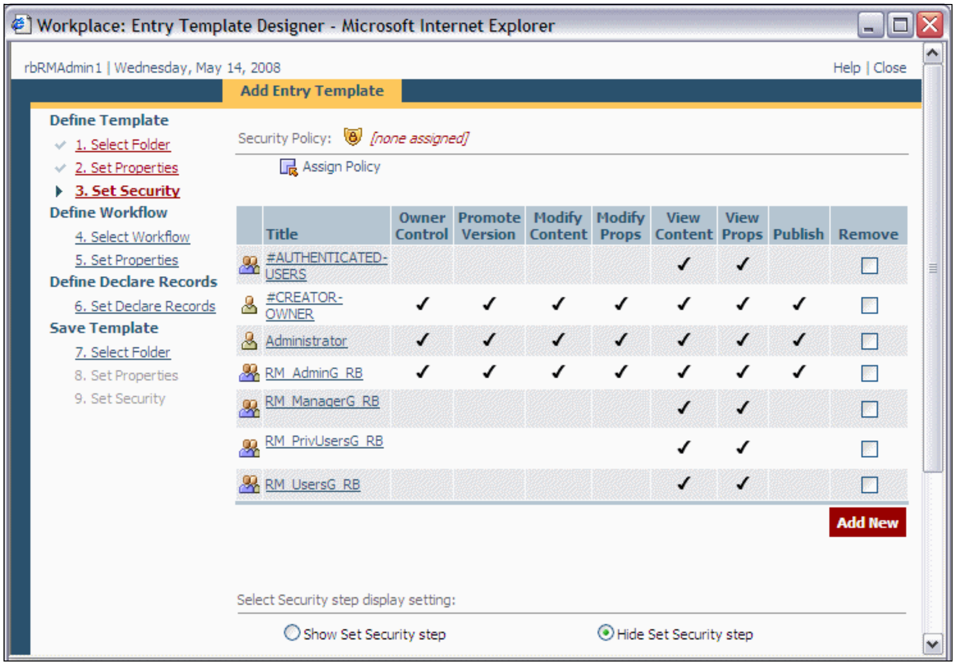
Figure 12-18 Entry Template: Select Properties

4. Set the security for the document entry template:

   a. Leave all default values as they appear.

   b. Select **Hide Set Security step**. Refer to Figure 12-19 on page 316.

   c. Click **Next**.



Workplace: Entry Template Designer - Microsoft Internet Explorer

rbRMAdmin1 | Wednesday, May 14, 2008 — Help | Close

Add Entry Template

Define Template
- ✓ 1. Select Folder
- ✓ 2. Set Properties
- ▸ 3. Set Security

Define Workflow
- 4. Select Workflow
- 5. Set Properties

Define Declare Records
- 6. Set Declare Records

Save Template
- 7. Select Folder
- 8. Set Properties
- 9. Set Security

Security Policy: [none assigned]

Assign Policy

Title	Owner Control	Promote Version	Modify Content	Modify Props	View Content	View Props	Publish	Remove
#AUTHENTICATED-USERS					✓	✓		□
#CREATOR-OWNER	✓	✓	✓	✓	✓	✓	✓	□
Administrator	✓	✓	✓	✓	✓	✓	✓	□
RM_AdminG_RB	✓	✓	✓	✓	✓	✓	✓	□
RM_ManagerG_RB					✓	✓		□
RM_PrivUsersG_RB					✓	✓		□
RM_UsersG_RB					✓	✓		□

Add New

Select Security step display setting:

○ Show Set Security step ● Hide Set Security step

Figure 12-19 Entry Template: Set Security

5. Click **Set Declare Records** on the left pane to move directly to Step 6 of the Template Designer, skipping the steps to define workflows, because we do not use workflow with this particular entry template.
6. Set the declare record options:
   a. Click **Browse/Search for Declare Record Template** to choose a Declare as Record Template that will be linked to this document entry template.
   b. Navigate to where your Declare as Record Template is located.

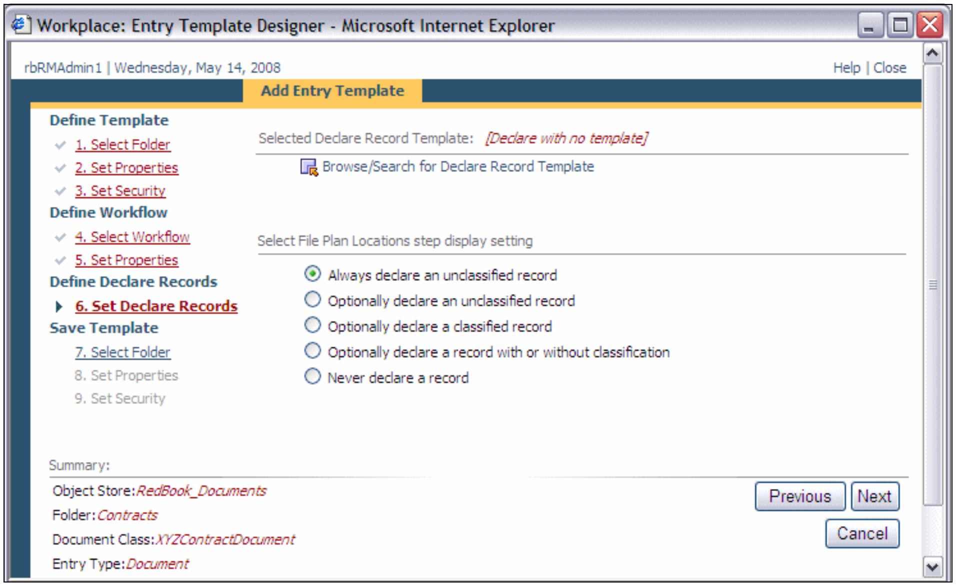
      For our case study, we navigate to **Redbook_Documents** → **Templates**.

   c. Select your Declare as Record Template.

      For our case study, we select **Declare Facility Management Contract Doc** template.

   d. Select **Always declare a record**.

      Note, the options here might vary depending on whether you have any Department of Defense (DoD) object stores configured in your environment. By choosing **Always declare a record**, you prevent users from having to make the decision. Refer to Figure 12-20 on page 317 for our case study setup.


Workplace: Entry Template Designer - Microsoft Internet Explorer
rbRMAdmin1 | Wednesday, May 14, 2008
Help | Close
Add Entry Template

Define Template
- ✓ 1. Select Folder
- ✓ 2. Set Properties
- ✓ 3. Set Security

Define Workflow
- ✓ 4. Select Workflow
- ✓ 5. Set Properties

Define Declare Records
- 6. Set Declare Records

Save Template
- 7. Select Folder
- 8. Set Properties
- 9. Set Security

Selected Declare Record Template: *[Declare with no template]*

Browse/Search for Declare Record Template

Select File Plan Locations step display setting

- ● Always declare an unclassified record
- ○ Optionally declare an unclassified record
- ○ Optionally declare a classified record
- ○ Optionally declare a record with or without classification
- ○ Never declare a record

Summary:

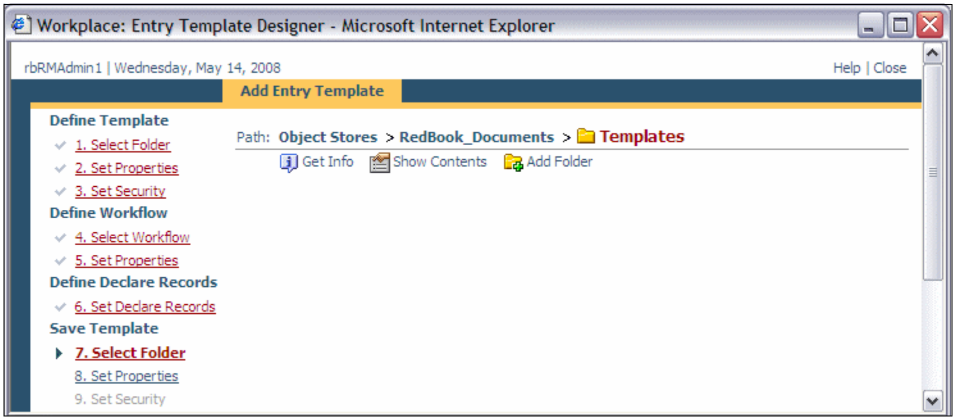
Object Store: *RedBook_Documents*
Folder: *Contracts*
Document Class: *XYZContractDocument*
Entry Type: *Document*

Figure 12-20 Entry Template: Set Declare Records option

e. Click **Next**.

7. Select the folder where you want to store the Document Entry Template, and click **Next**.

   For our case, we navigate to **Redbook_Documents** → **Templates** (refer to Figure 12-21).


Workplace: Entry Template Designer - Microsoft Internet Explorer
rbRMAdmin1 | Wednesday, May 14, 2008
Help | Close
Add Entry Template

Define Template
- ✓ 1. Select Folder
- ✓ 2. Set Properties
- ✓ 3. Set Security

Define Workflow
- ✓ 4. Select Workflow
- ✓ 5. Set Properties

Define Declare Records
- ✓ 6. Set Declare Records

Save Template
- 7. Select Folder
- 8. Set Properties
- 9. Set Security

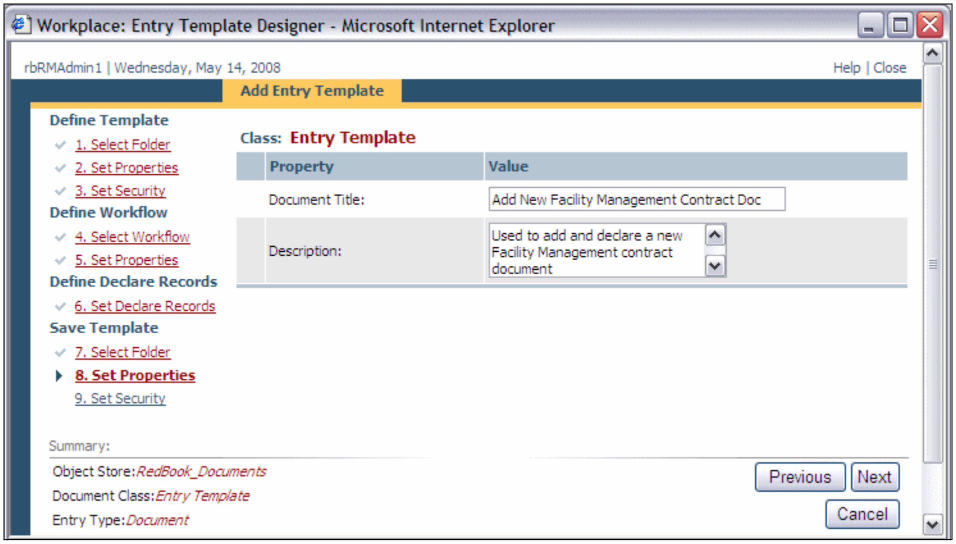
Path: **Object Stores** > **RedBook_Documents** > **Templates**

Get Info
Show Contents
Add Folder

Figure 12-21 Entry Template: Select Folder (for saved entry template)

8. Enter the document entry template properties, and click **Next**.

   For our case study, we enter Add New Facility Management Contract Doc. Refer to Figure 12-22.



Workplace: Entry Template Designer - Microsoft Internet Explorer

rbRMAdmin1 | Wednesday, May 14, 2008

Help | Close

Add Entry Template

Define Template
- ✓ 1. Select Folder
- ✓ 2. Set Properties
- ✓ 3. Set Security

Define Workflow
- ✓ 4. Select Workflow
- ✓ 5. Set Properties

Define Declare Records
- ✓ 6. Set Declare Records

Save Template
- ✓ 7. Select Folder
- ▶ 8. Set Properties
- 9. Set Security

Class: Entry Template

Property	Value
Document Title:	Add New Facility Management Contract Doc
Description:	Used to add and declare a new Facility Management contract document

Summary:

Object Store: *RedBook_Documents*

Document Class: *Entry Template*

Entry Type: *Document*

Figure 12-22 Entry Template: Set Properties (for saved entry template)

9. Set the security permissions to determine which users can access this entry template.

   For our case study, we leave the values as they appear.

10. Click **Finish**, and then, click **OK** to acknowledge the confirmation.

The document entry template is now saved and can be used from Workplace to add a new Facility Management Contract document.

12.4 Declare a record using the entry template

The combined document entry template and Declare as Record Template can now be tested. To test the templates, complete the following steps:

1. From Workplace, navigate to the document entry template.
2. Select the appropriate entry template for the type of document that we intend to add.
3. Enter the required property values for the document.

4. Browse the file system to locate the actual document to add.
5. Complete the add/declare.

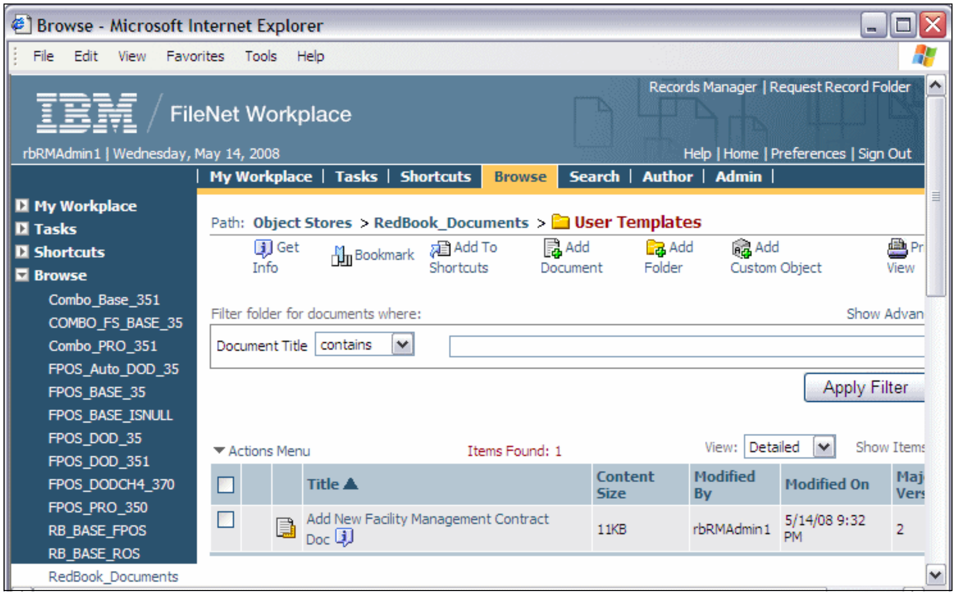
The detailed instructions are:

1. In Workplace, browse to where you store your document entry template.

   For our case study, we select **Browse** → **Redbook_Documents** → **User Templates**.

2. Select your document entry template.

   For our case study, we select **Add New Facility Management Contract Doc**. Refer to Figure 12-23.

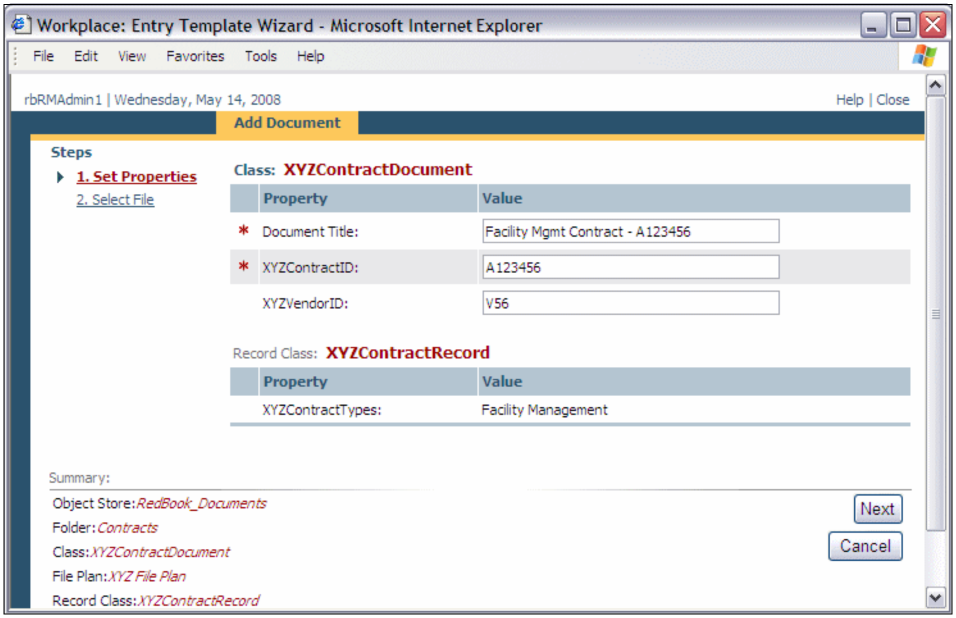


			Title ▲	Content Size	Modified By	Modified On	Maj Vers
□			Add New Facility Management Contract Doc	11KB	rbRMAdmin1	5/14/08 9:32 PM	2

Figure 12-23 Validate: Select document entry template to create a document

3. Set the property values for the new document.

   Figure 12-24 on page 320 shows the values that we enter for a facility management document.


Workplace: Entry Template Wizard - Microsoft Internet Explorer
File Edit View Favorites Tools Help
rbRMAdmin1 | Wednesday, May 14, 2008
Help | Close
Add Document
Steps
1. Set Properties
2. Select File
Next
Cancel

Class: XYZContractDocument

	Property	Value
*	Document Title:	Facility Mgmt Contract - A123456
*	XYZContractID:	A123456
	XYZVendorID:	V56

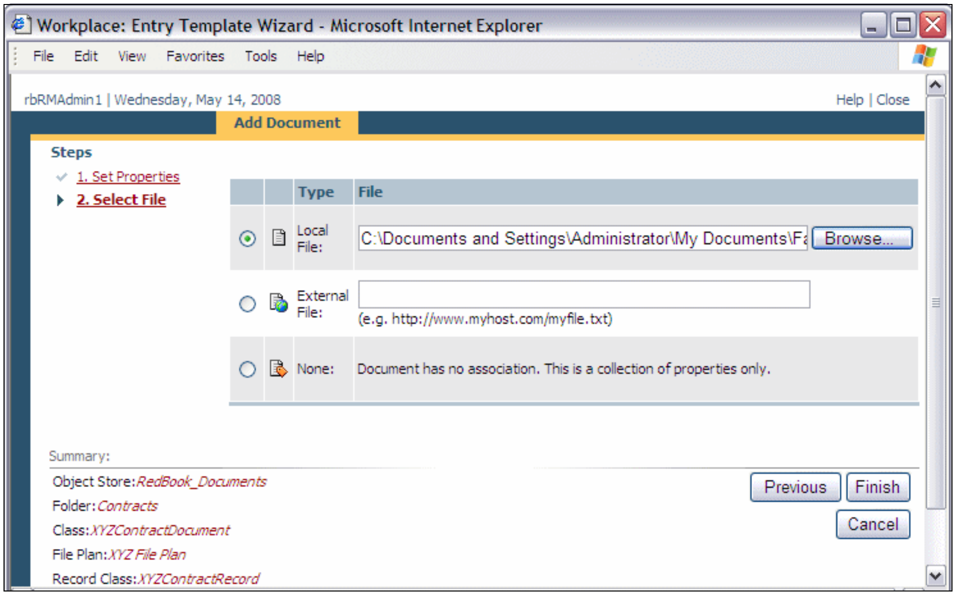
Record Class: XYZContractRecord

	Property	Value
	XYZContractTypes:	Facility Management

Summary:
Object Store: *RedBook_Documents*
Folder: *Contracts*
Class: *XYZContractDocument*
File Plan: *XYZ File Plan*
Record Class: *XYZContractRecord*

Figure 12-24 Validate: Enter document properties

4. Click **Next**.
5. Click **Browse** to locate and select the actual document to add to the system. Figure 12-25 on page 321 shows that a local file is selected.



rbRMAdmin1 | Wednesday, May 14, 2008

Steps

✓ 1. Set Properties

▸ 2. Select File

		Type	File
●		Local File:	C:\Documents and Settings\Administrator\My Documents\F
○		External File:	(e.g. http://www.myhost.com/myfile.txt)
○		None:	Document has no association. This is a collection of properties only.

Summary:

Object Store: *RedBook_Documents*

Folder: *Contracts*

Class: *XYZContractDocument*

File Plan: *XYZ File Plan*

Record Class: *XYZContractRecord*

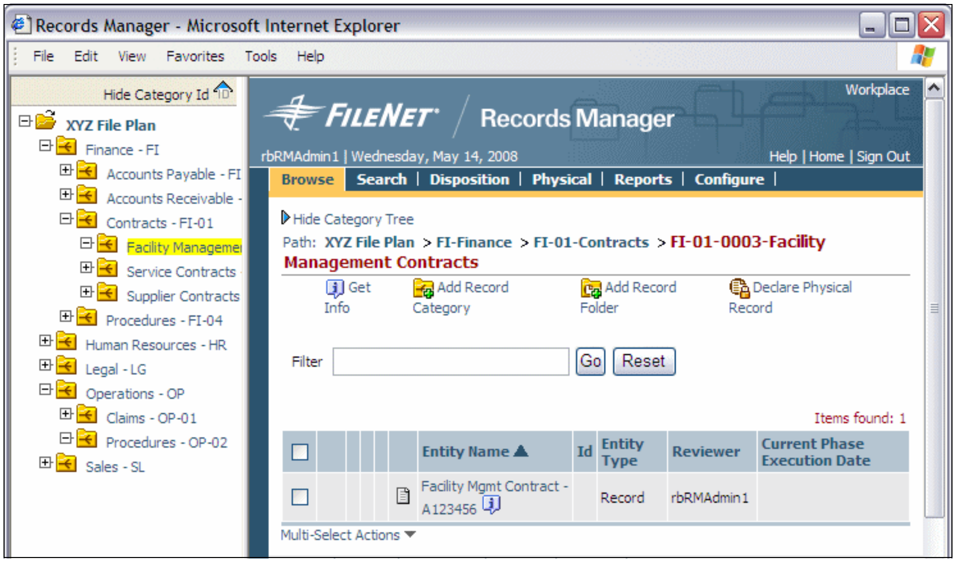
Figure 12-25 Validate: Add a file to the document

6. Click **Finish**, and then, click **OK** to finish adding the document.

To validate that the document is declared as a record, perform the following steps:

1. Launch IBM FileNet Records Manager.
2. Select **Browse** and navigate to the file plan location where the record appears.

   For our case study, we navigate to **XYZ File Plan** → **Finance** → **Contracts** → **Facility Management Contracts**. Refer to Figure 12-26 on page 322.



Path: XYZ File Plan > FI-Finance > FI-01-Contracts > FI-0003-Facility Management Contracts

Items found: 1

□	Entity Name ▲	Id	Entity Type	Reviewer	Current Phase Execution Date
□	Facility Mgmt Contract - A123456		Record	rbRMAdmin1	

Figure 12-26 Validate: Browse to see the new record in the file plan

Notice that a facility management record has been added to the file plan.

12.5 Building templates for other contract types

To complete the scenario for the case study, we repeat the preceding tasks for both Service contracts and Supplier contracts. We need one declare as record template and one document entry template for each additional contract type. A user who wants to add new documents then chooses the appropriate document entry template based on the type of contract document that is being added, and the system automatically declares the document as a record in the corresponding file plan component.

13

Records disposition case study

In Chapter 6, “Records disposition” on page 149, we discussed the concept of disposition. In this chapter, we show you the required steps to perform disposition-related activities.

We describe the following topics in this chapter:

- Case study disposition scenarios
- Creating a disposition schedule:
  - Adding an action
  - Adding an event trigger
  - Adding the disposition schedule
- Assigning the disposition schedule to an existing record entity

  Searching for disposition schedules assigned to record entities.
- Running the Disposition Sweep:
  - Configuring the Disposition Sweep
  - Executing the Disposition Sweep
- Initiating disposition on an item that is eligible for disposition:
  - Initiating disposition on entities
  - Processing items

13.1 Case study disposition scenarios

Our case study is based on the fictitious Fictional Insurance Company X that we introduced in 3.3.2, “Example file plan” on page 65. The company’s file plan is shown in Figure 3-3 on page 66. The partial file plan showing the categories that are related to the case study for this book is illustrated in Figure 13-1.

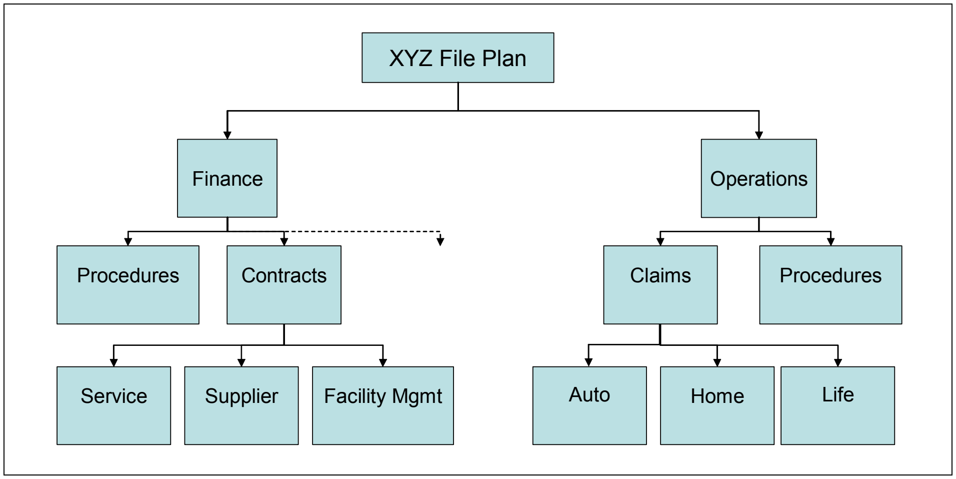


Figure 13-1 Partial file plan showing categories related to the case study for this book

Depending on litigation or business practices, your company might be asked to put specific records on hold. For our case study, we use the disposition schedule that is s described in Table 13-1 on page 325.

Table 13-1 Partial retention schedule showing the case study examples

Record series	Description	Total retention period	Disposition action
Claims	Insurance claim documents	Claim Closed + 5 Years	Destroy
Procedures	Internal corporate procedures for any business unit	Superceded + 10 Years	Destroy
Facility Management Contracts	Contract documents pertaining to facility management contracts	Contract Expiration + 3 Years	Destroy
Service Contracts	Contract documents pertaining to service contracts	Contract Expiration + 7 Years	Destroy
Supplier Contracts	Contract documents pertaining to supplier contracts	Contract Expiration + 7 Years	Destroy

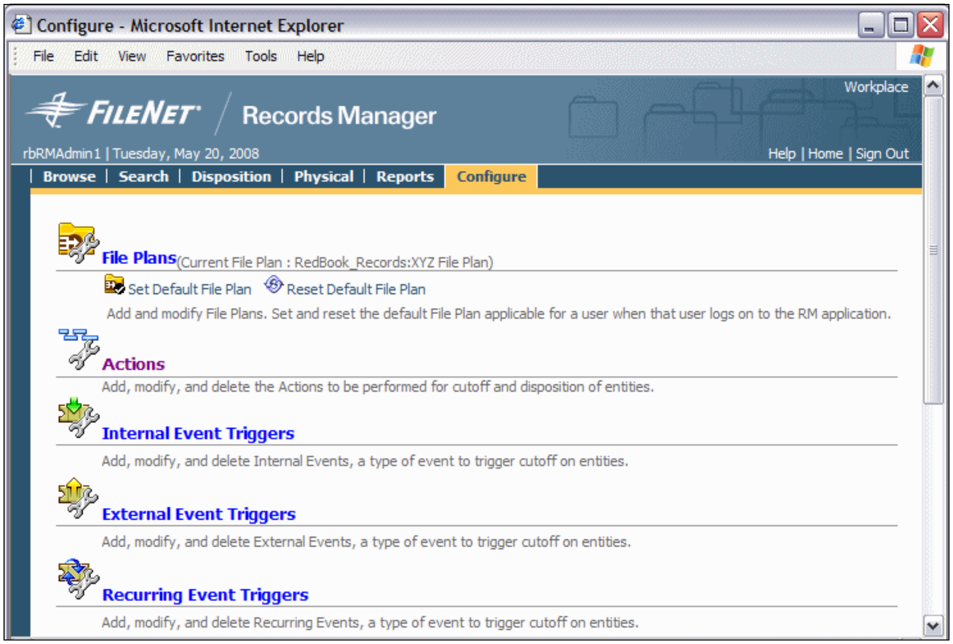
13.2 Defining a disposition schedule

To define a disposition schedule, you first must add actions and event triggers and then create the disposition schedules.

13.2.1 Adding an action

For our case study, we need to add a Destroy Action. To add an action, follow these steps:

1. Log on to the IBM FileNet Records Manager application.
2. Select the **Configure** tab. Figure 13-2 on page 326 appears.


Configure - Microsoft Internet Explorer
File Edit View Favorites Tools Help
FileNet Records Manager
Workplace
rbRMAdmin1 | Tuesday, May 20, 2008
Help | Home | Sign Out
Browse | Search | Disposition | Physical | Reports | Configure

File Plans(Current File Plan : RedBook_Records:XYZ File Plan)

Set Default File Plan Reset Default File Plan

Add and modify File Plans. Set and reset the default File Plan applicable for a user when that user logs on to the RM application.

Actions

Add, modify, and delete the Actions to be performed for cutoff and disposition of entities.

Internal Event Triggers

Add, modify, and delete Internal Events, a type of event to trigger cutoff on entities.

External Event Triggers

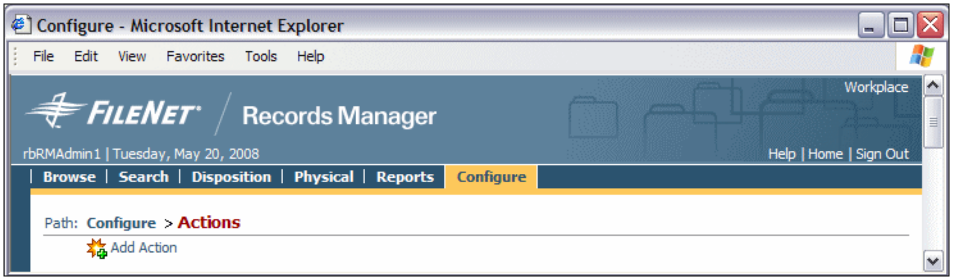
Add, modify, and delete External Events, a type of event to trigger cutoff on entities.

Recurring Event Triggers

Add, modify, and delete Recurring Events, a type of event to trigger cutoff on entities.

Figure 13-2 Configure Actions

3. Click **Actions**. Figure 13-3 appears.

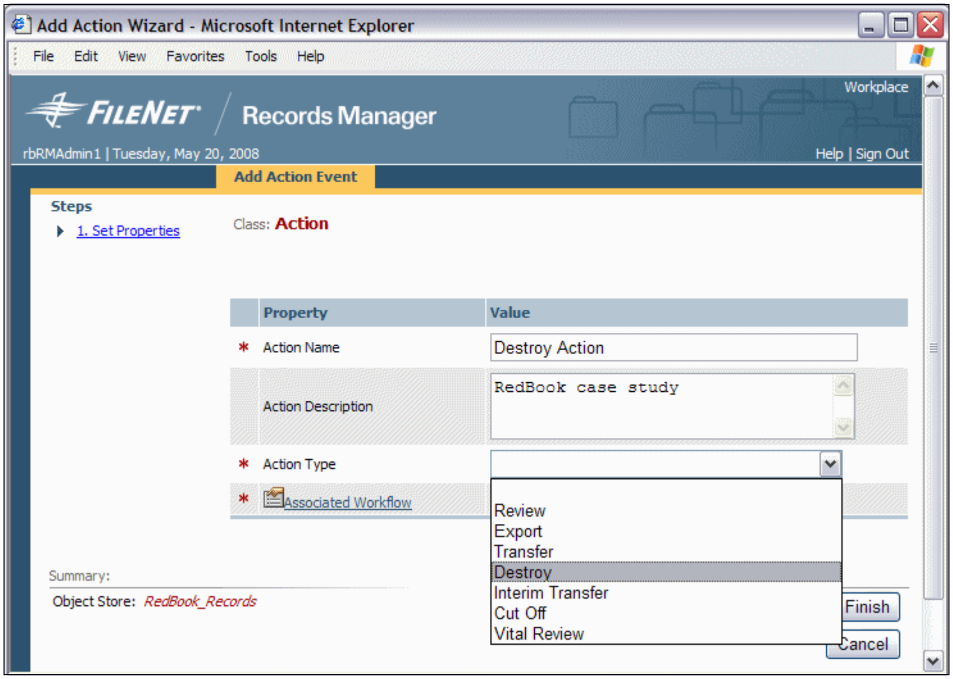

Configure - Microsoft Internet Explorer
File Edit View Favorites Tools Help
FileNet Records Manager
Workplace
rbRMAdmin1 | Tuesday, May 20, 2008
Help | Home | Sign Out
Browse | Search | Disposition | Physical | Reports | Configure

Path: Configure > Actions

Add Action

Figure 13-3 Configure: Add Action

4. Click **Add Action**. The Add Action Event window appears. Refer to Figure 13-4 on page 327.
5. Enter the action name (up to 20 characters), description, and select the action type. Existing action types include: Cut Off, Destroy, Export, Interim Transfer, Review, Transfer, or Vital Review. You can also create your own action type. For our case study, we select **Destroy** as the action type. Refer to our setup in Figure 13-4 on page 327.



Class: Action

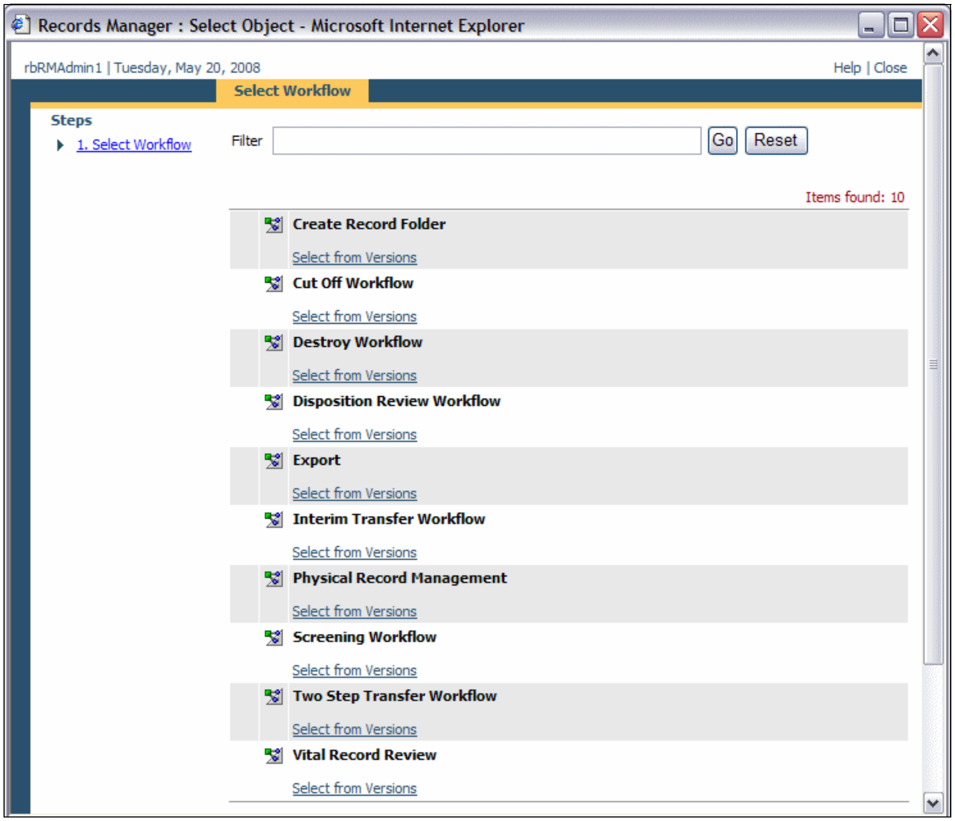
Property	Value
* Action Name	Destroy Action
Action Description	RedBook case study
* Action Type	
* Associated Workflow	

Object Store: *RedBook_Records*

Figure 13-4 Add Action Event: Select Action Type

6. Select a workflow to be launched for this action. You can select either one of the standard workflows that are provided or create your own workflow:

   a. Click **Associated Workflow**. A list of workflows appears. Refer to Figure 13-5 on page 328.



Records Manager : Select Object - Microsoft Internet Explorer

rbRMAdmin1 | Tuesday, May 20, 2008

Select Workflow

Steps

1. Select Workflow

Filter

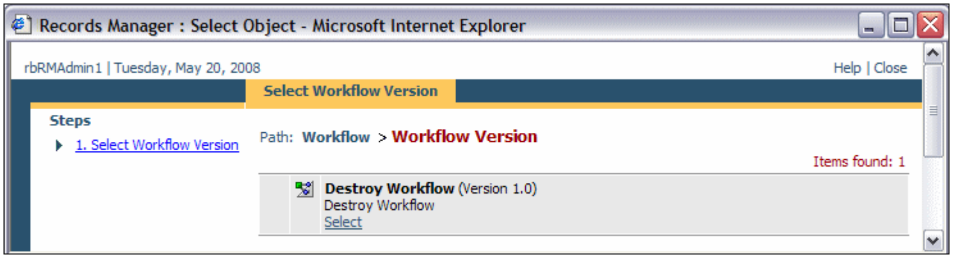
Items found: 10

- Create Record Folder
  Select from Versions
- Cut Off Workflow
  Select from Versions
- Destroy Workflow
  Select from Versions
- Disposition Review Workflow
  Select from Versions
- Export
  Select from Versions
- Interim Transfer Workflow
  Select from Versions
- Physical Record Management
  Select from Versions
- Screening Workflow
  Select from Versions
- Two Step Transfer Workflow
  Select from Versions
- Vital Record Review
  Select from Versions

Figure 13-5 Add Action Event: Associated Workflow selection list

b. Select the **Select from Versions** link under the associated workflow that you want for your action event. For our case study, we select the **Select from Versions** link under Destroy Workflow.

c. Click the **Select** link under the appropriate workflow version. For our case study, there is only one version of the Destroy Workflow. Refer to Figure 13-6.



Records Manager : Select Object - Microsoft Internet Explorer

rbRMAdmin1 | Tuesday, May 20, 2008

Select Workflow Version

Steps

1. Select Workflow Version

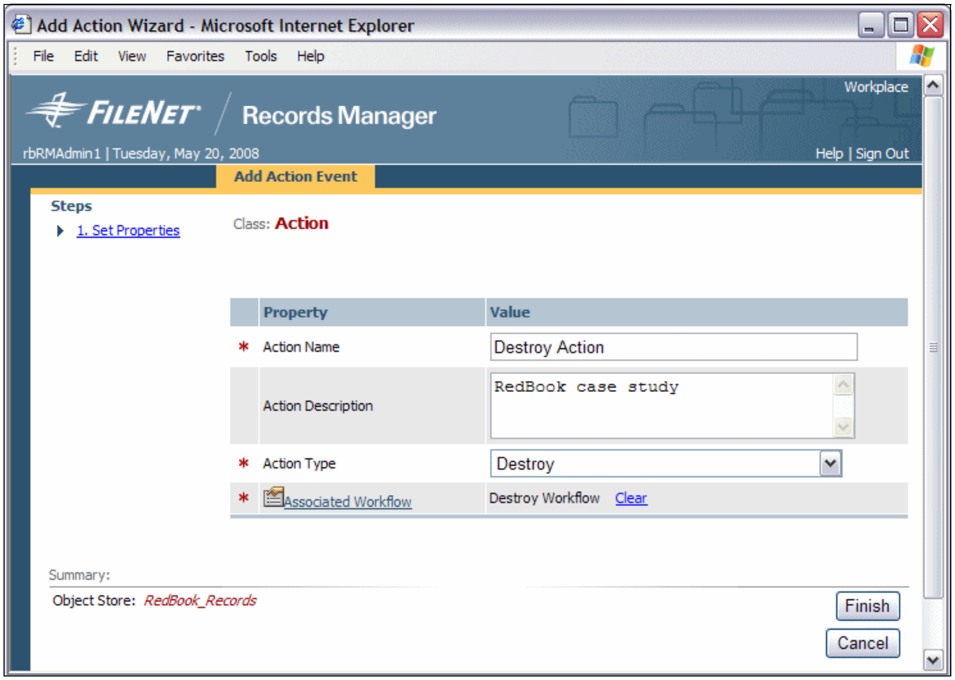
Path: Workflow > Workflow Version

Items found: 1

Destroy Workflow (Version 1.0)
Destroy Workflow
Select

Figure 13-6 Action Action Event: Select Destroy Workflow version

7. Click **Finish** at the completed action event window (refer to Figure 13-7).


Add Action Wizard - Microsoft Internet Explorer
File Edit View Favorites Tools Help
FILENET Records Manager
Workplace
rbRMAdmin1 | Tuesday, May 20, 2008
Help | Sign Out
Add Action Event
Steps
1. Set Properties
Finish
Cancel

Class: Action

	Property	Value
*	Action Name	Destroy Action
	Action Description	RedBook case study
*	Action Type	Destroy
*	Associated Workflow	Destroy Workflow Clear

Summary:

Object Store: RedBook_Records

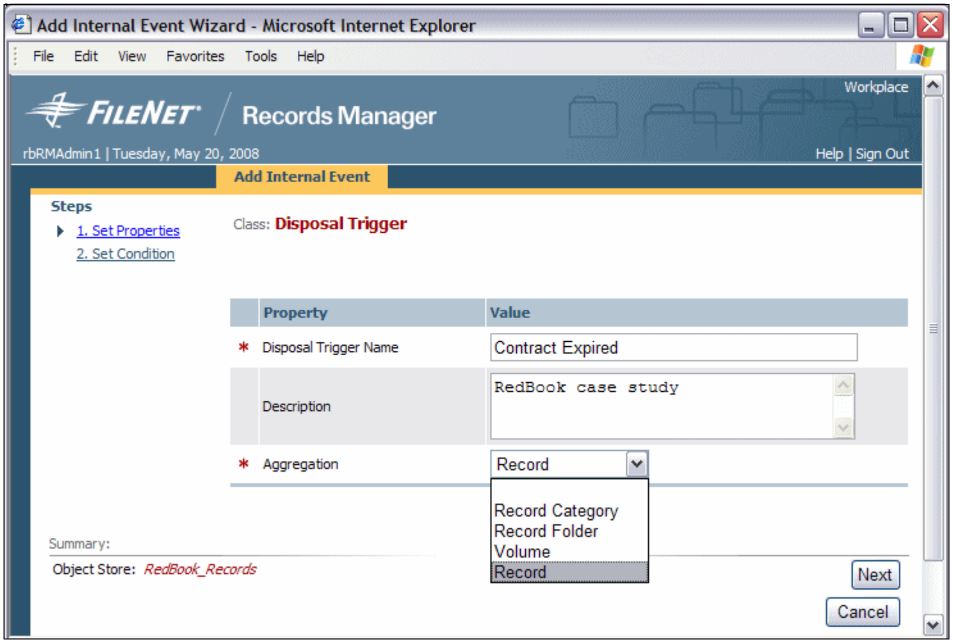
Figure 13-7 Add Action Event: Destroy Action

8. Click **OK** to complete the addition of the action event.

13.2.2 Adding an event trigger

For our case study, we need to add an action trigger called Contract Expired. To add the event trigger, follow these steps for each internal event trigger:

1. Select the **Configure** tab (continue from the previous step).
2. Click **Internal Event Triggers**.
3. Click **Add Internal Event**.
4. Set the properties for the event trigger by entering the trigger name (up to 20 characters) and trigger description and select the trigger aggregation type. Existing trigger aggregation types include Record Category, Record Folder, Volume, and Record. For our case study, we select **Record** as the trigger aggregation type. Refer to our setup in Figure 13-8 on page 330.



Class: Disposal Trigger

Property	Value
* Disposal Trigger Name	Contract Expired
Description	RedBook case study
* Aggregation	Record

Summary:

Object Store: *RedBook_Records*

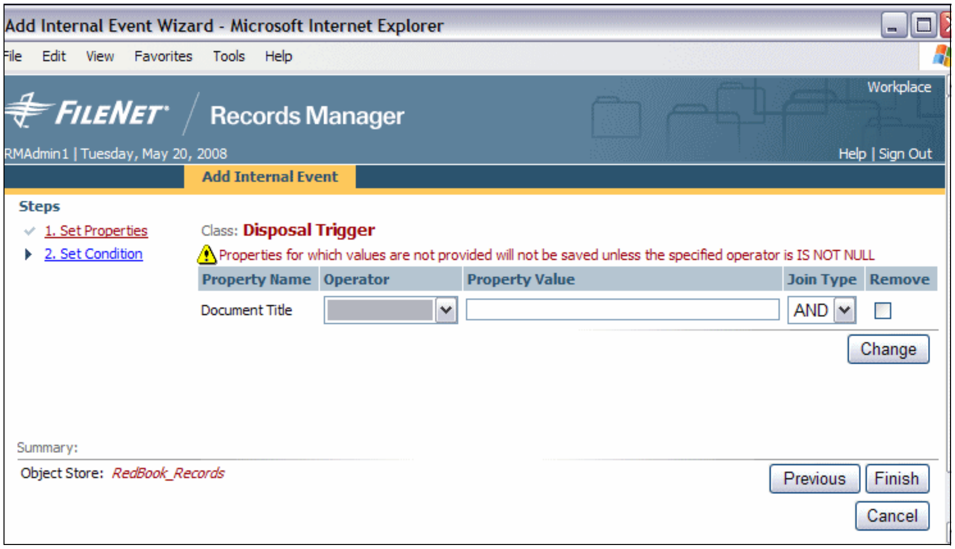
Figure 13-8 Add Internal Event Trigger: Set Properties

5. Set the condition of the trigger. Conditions define the triggering criteria for an internal event. You can define up to five conditions for a single trigger.

   For our case study, we want the event to trigger when a user or a certain process provides an expiration date for a contract. In other words, we want to set the trigger's condition to when the value of the property XYZContractExpirationDate is no longer null.

   To set this condition (XYZContractExpirationDate is not null):

   a. Click **Next** or the **Set Condition** link on the left pane. Figure 13-9 on page 331 appears.



Class: **Disposal Trigger**

⚠ Properties for which values are not provided will not be saved unless the specified operator is IS NOT NULL

Property Name	Operator	Property Value	Join Type	Remove
Document Title			AND	□

Summary:

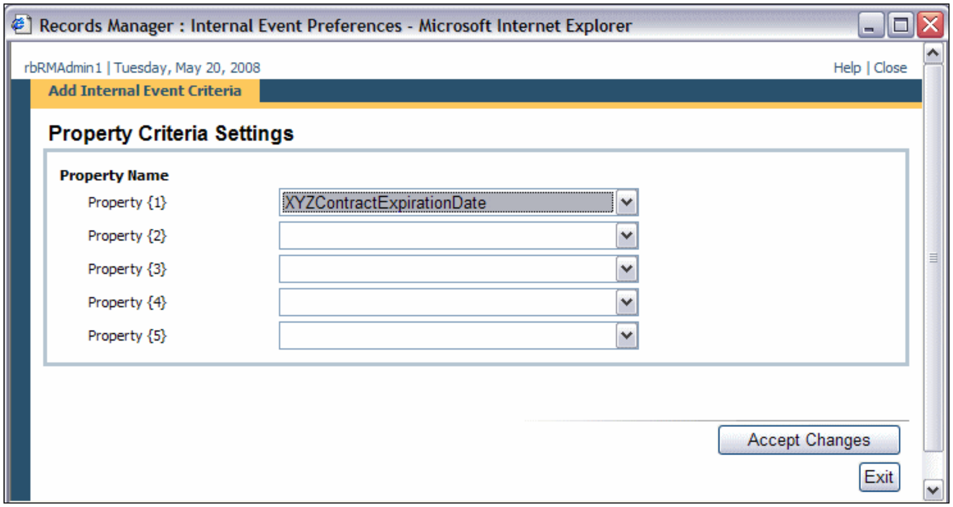
Object Store: *RedBook_Records*

Figure 13-9 Add Internal Event Trigger: Set Condition

b. Select the appropriate property name by following these steps:

i. Click **Change**.

ii. From each property's drop-down menu, select the property that you want to use for the trigger. For our case study, we select **XYZContractExpirationDate**. Refer to Figure 13-10.


Records Manager : Internal Event Preferences - Microsoft Internet Explorer
rbRMAdmin1 | Tuesday, May 20, 2008
Help | Close
Add Internal Event Criteria
Accept Changes
Exit

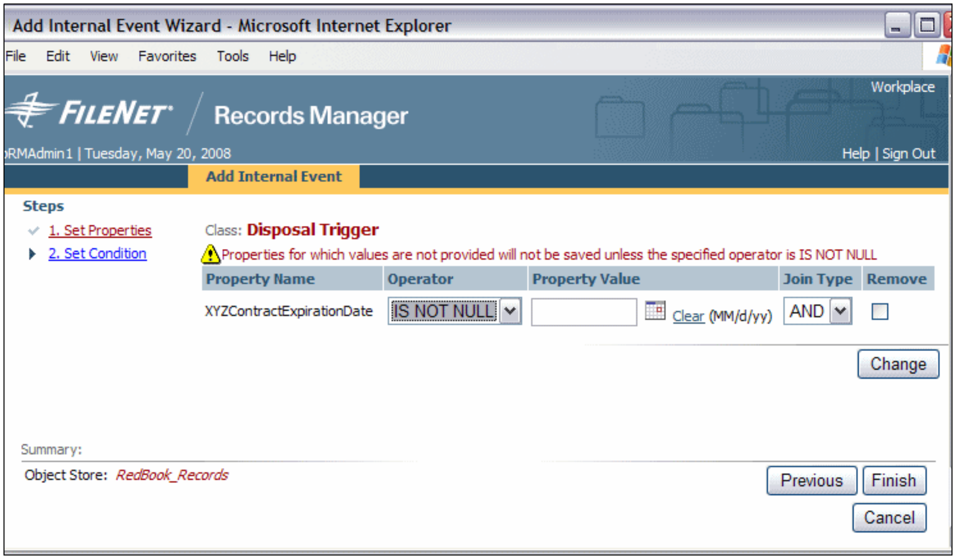
Property Criteria Settings

Property Name	
Property {1}	XYZContractExpirationDate
Property {2}	
Property {3}	
Property {4}	
Property {5}	

Figure 13-10 Add Internal Event Trigger: Set Property Criteria Settings

iii. Click **Accept Changes**.

c. From the Operator drop-down menu, select **IS NOT NULL**. Refer to Figure 13-11.



Add Internal Event Wizard - Microsoft Internet Explorer

RMAdmin1 | Tuesday, May 20, 2008

Class: Disposal Trigger

Properties for which values are not provided will not be saved unless the specified operator is IS NOT NULL

Property Name	Operator	Property Value	Join Type	Remove
XYZContractExpirationDate	IS NOT NULL	Clear (MM/d/yy)	AND	□

Summary:

Object Store: *RedBook_Records*

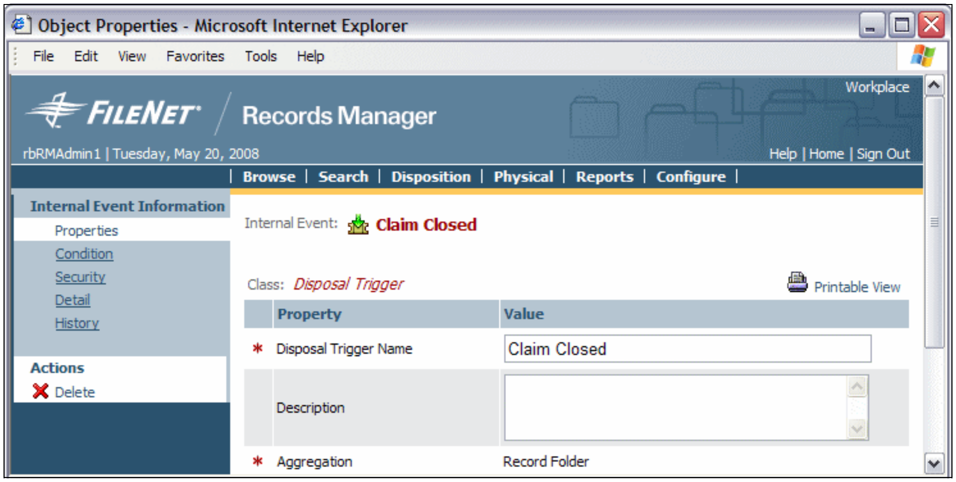
Figure 13-11 Add Internal Event Trigger: Set Condition completed

6. Click **Finish**, and then, click **OK** on the confirmation page.

For our case study, we create another internal event trigger called Claim Closed. This event triggers at 5 years after a claim is closed. The trigger summary is shown in Table 13-2. Refer to Figure 13-12 on page 333 and Figure 13-13 on page 333 for the setup in the system.

Table 13-2 Claim Closed trigger information

Field information	**Value**
Disposal Trigger Name	Claim Closed
Description	RedBook case study
Aggregation	Record Folder
Condition	XYZClaimClosedDate IS NOT NULL

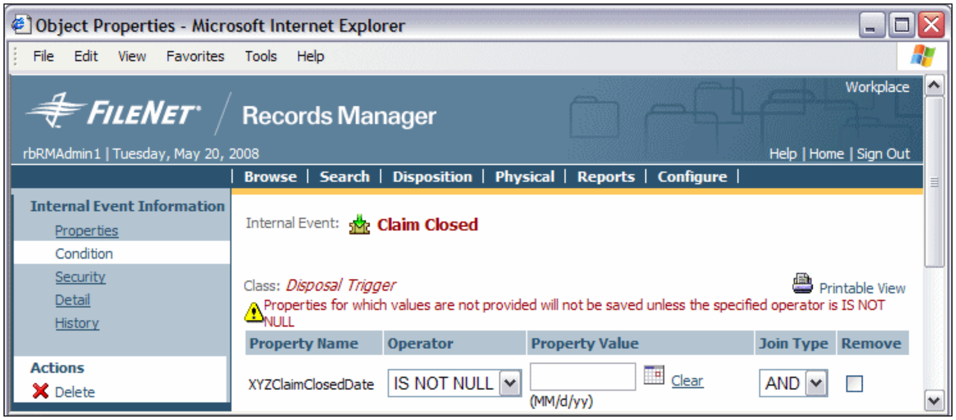

Object Properties - Microsoft Internet Explorer
File Edit View Favorites Tools Help
FileNet Records Manager
Workplace
rbRMAdmin1 | Tuesday, May 20, 2008
Help | Home | Sign Out
Browse | Search | Disposition | Physical | Reports | Configure
Internal Event Information
Properties
Condition
Security
Detail
History
Actions
Delete
Printable View

Internal Event: Claim Closed

Class: *Disposal Trigger*

	Property	Value
*	Disposal Trigger Name	Claim Closed
	Description	
*	Aggregation	Record Folder

Figure 13-12 Internal Event Trigger: Claim Closed: Properties


Object Properties - Microsoft Internet Explorer
File Edit View Favorites Tools Help
FileNet Records Manager
Workplace
rbRMAdmin1 | Tuesday, May 20, 2008
Help | Home | Sign Out
Browse | Search | Disposition | Physical | Reports | Configure
Internal Event Information
Properties
Condition
Security
Detail
History
Actions
Delete
Printable View

Internal Event: Claim Closed

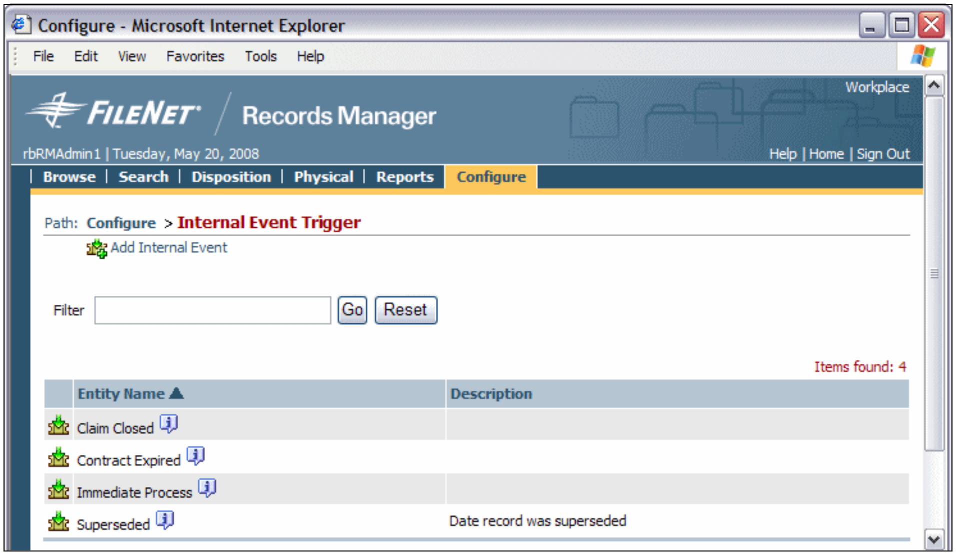
Class: *Disposal Trigger*

Properties for which values are not provided will not be saved unless the specified operator is IS NOT NULL

Property Name	Operator	Property Value	Join Type	Remove
XYZClaimClosedDate	IS NOT NULL	(MM/d/yy) Clear	AND	□

Figure 13-13 Internal Event Trigger: Claim Closed: Condition

Figure 13-14 on page 334 shows a list of the internal trigger events that we have created for our case study. To browse to the detailed internal trigger event window, click the **Info Icon** next to the trigger. You can click the **Condition** link on the left pane to see the details of the condition that has been set up for the trigger or you can click **Detail** to see the entire detailed setup for the trigger.



Configure - Microsoft Internet Explorer
File Edit View Favorites Tools Help
FILENET Records Manager
Workplace
rbRMAdmin1 | Tuesday, May 20, 2008
Help | Home | Sign Out
Browse | Search | Disposition | Physical | Reports | Configure
Add Internal Event
Filter
Go
Reset

Path: Configure > Internal Event Trigger

Items found: 4

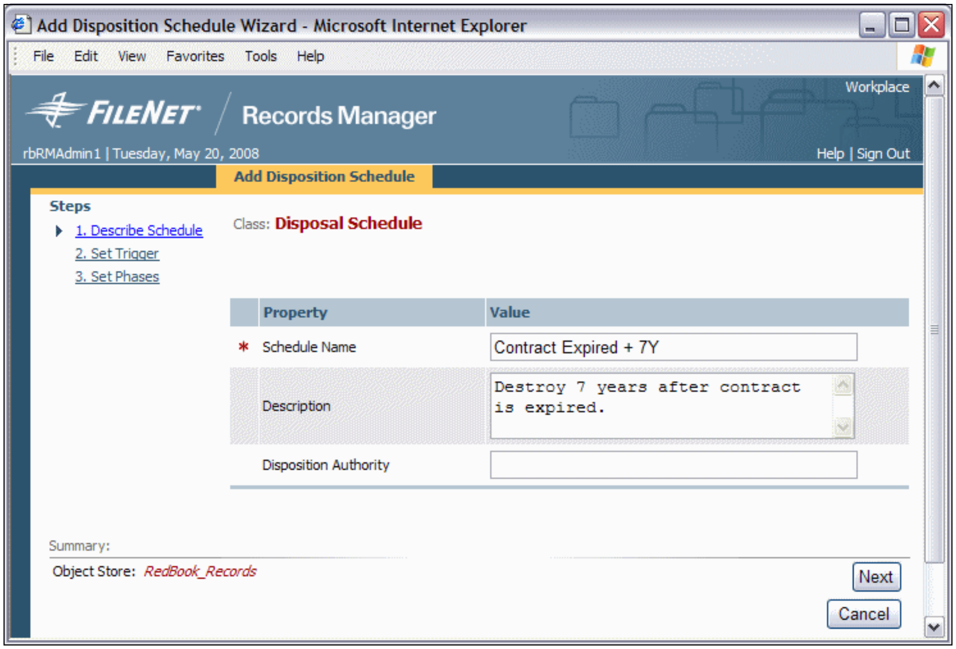
Entity Name	Description
Claim Closed	
Contract Expired	
Immediate Process	
Superseded	Date record was superseded

Figure 13-14 Configure: Internal Event Triggers

13.2.3 Adding the disposition schedule

After you define the action and event triggers to be used in a disposition schedule, you can now add the disposition schedule to the system by following these steps:

1. Select the **Disposition** tab.
2. Click **Disposition Schedules**.
3. Click **Add Disposition Schedule**.
4. Enter the disposition schedule name, description, and the name of the authority responsible for the disposition of the entities. For our case study, the disposition schedule is called Contract Expired + 7Y. Figure 13-15 on page 335 shows the setup of the disposition schedule.



Class: Disposal Schedule

	Property	Value
*	Schedule Name	Contract Expired + 7Y
	Description	Destroy 7 years after contract is expired.
	Disposition Authority	

Summary:
Object Store: *RedBook_Records*

Figure 13-15 Add Disposition Schedule: Describe Schedule

5. Set the trigger for the disposition schedule:

   a. Click **Next**.

   b. Select the trigger type and the trigger that is used in the disposition schedule:

      i. For our case study, we select **Internal Event**.

      ii. From the Internal Event drop-down list box, we select the event that we created in the earlier section, **Contract Expired**.

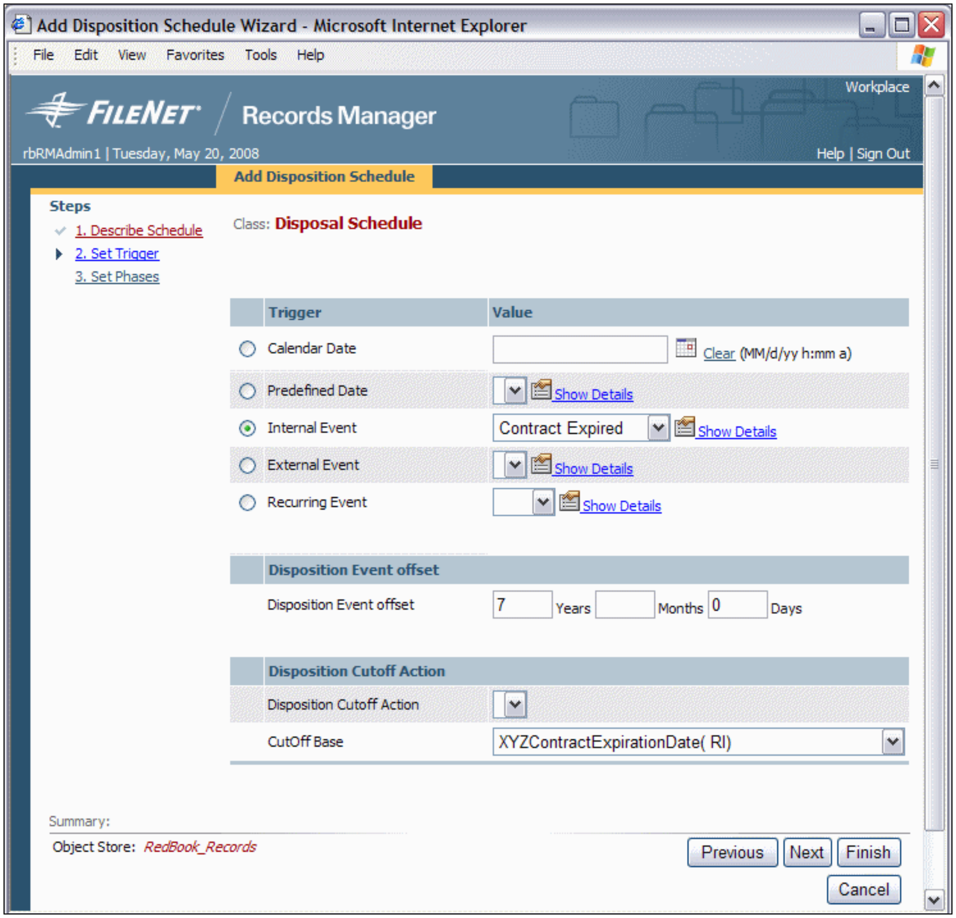
   c. Enter the disposition event offset in years, months, and days. This offset specifies a time interval between the start of the cutoff and the launch of the associated cutoff action.

      For our case study, when the contract expiration date is set (no longer null), the cutoff starts. However, we must wait 7 years after the contract is expired before any associated cutoff action can be launched. Thus, our offset is 7 years. We enter 7 for event offset years.

   d. Select the cutoff action and the cutoff base for the disposition. The CutOff Base drop-down list provides a set of IBM FileNet Records Manager date properties that you can use to calculate cutoff dates based on individual IBM FileNet Records Manager entity properties. If you leave this set to [Event Date], Disposition Sweep uses the trigger event date for the cutoff calculation. Disposition Sweep adds the disposition event offset to the cutoff base date to compute the cutoff date.

For our case study, our cutoff is based on the contract expiration date. We leave the Cutoff Action field blank. For the Cutoff Base field, we select **XYZContractExpirationDate**.

Figure 13-16 shows the trigger setup for our case study disposition schedule.



Class: **Disposal Schedule**

	Trigger	Value
○	Calendar Date	Clear (MM/d/yy h:mm a)
○	Predefined Date	Show Details
●	Internal Event	Contract Expired Show Details
○	External Event	Show Details
○	Recurring Event	Show Details

Disposition Event offset

Disposition Event offset: 7 Years, Months, 0 Days

Disposition Cutoff Action

Disposition Cutoff Action:

CutOff Base: XYZContractExpirationDate(RI)

Summary:

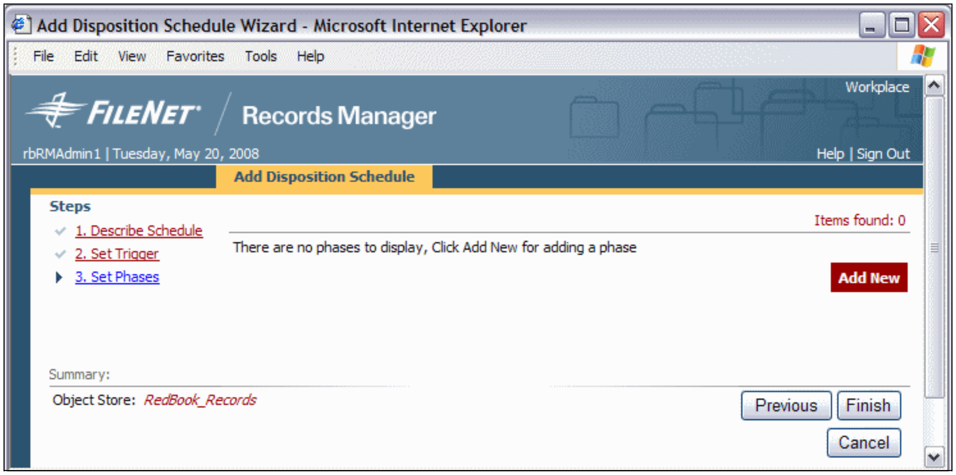
Object Store: *RedBook_Records*

Figure 13-16 Add Disposition Schedule: Set Trigger

6. Set the phases for the disposition schedule.

   You can set one or more phases for a disposition schedule. Each phase specifies a disposition action and the retention period for an entity. For our case study, we want to immediately destroy the records 7 years after a contract is expired. We want to have 0 days of retention period. In this case, we need one phase that is set to 0 day. Follow these steps to set the phase:

a. Click **Next**, or click **Set Phases**. Figure 13-17 appears.



Items found: 0

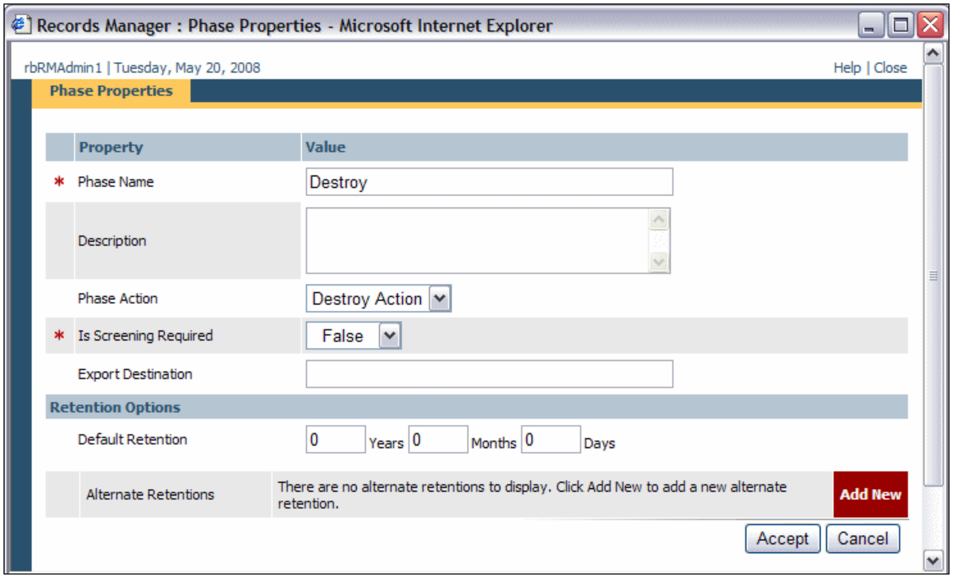
There are no phases to display, Click Add New for adding a phase

Summary:

Object Store: RedBook_Records

Figure 13-17 Add Disposition Schedule: Set Phases initial page

b. Click **Add New** to open the Phase Properties page. Refer to Figure 13-18.



Records Manager : Phase Properties - Microsoft Internet Explorer
rbRMAdmin1 | Tuesday, May 20, 2008
Help | Close
Phase Properties
Add New
Accept
Cancel

	Property	Value
*	Phase Name	Destroy
	Description	
	Phase Action	Destroy Action
*	Is Screening Required	False
	Export Destination	

Retention Options

Default Retention	0 Years 0 Months 0 Days
Alternate Retentions	There are no alternate retentions to display. Click Add New to add a new alternate retention.

Figure 13-18 Add Disposition Schedule: Set Phase Properties

c. Enter a name and description for the phase.

   For our case study, we enter Destroy.

d. From the Phase Action drop-down menu, select an action that you have created.

   For our case study, we select **Destroy Action**.

e. For the “Is Screening Required” field, select **True** if you want to review records before launching the workflow associated with the phase. If screening is not required, select **False**.

   For our case study, we do not need screening. We select **False**.

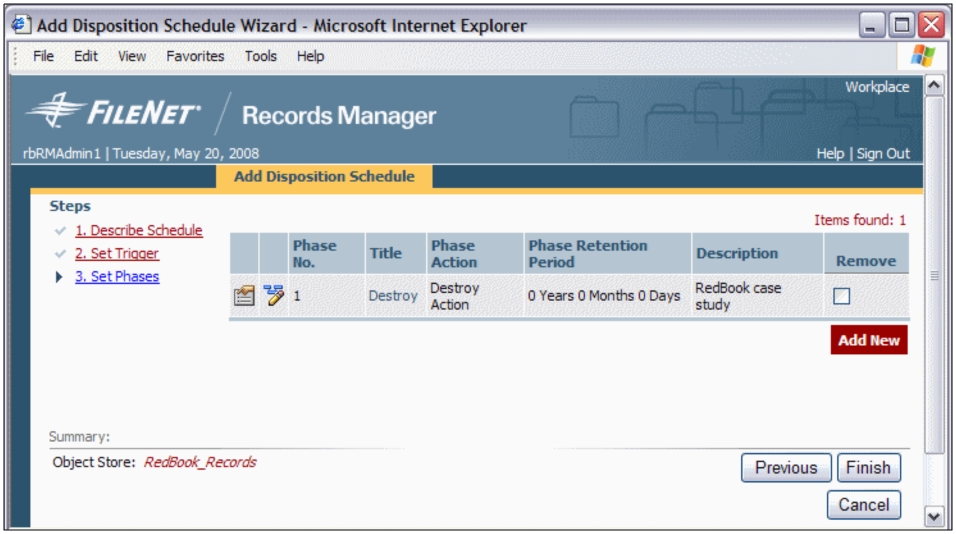
f. Specify a Default Retention period for entities in this phase. A value is required if you are not assigning alternate retention periods.

   For our case study, we enter 0 Years, 0 Months, 0 Days.

   Figure 13-18 on page 337 shows the phase setup page for our case study.

g. Click **Accept**.

Figure 13-19 shows the current setup for our case study with the phase information.



		Phase No.	Title	Phase Action	Phase Retention Period	Description	Remove
		1	Destroy	Destroy Action	0 Years 0 Months 0 Days	RedBook case study	□

Summary:

Object Store: *RedBook_Records*

Figure 13-19 Add Disposition Schedule: Set Phases completed

7. Click **Finish**, and then, click **OK** on the confirmation page.

For our case study, we need to create two more disposition schedules as defined in Table 13-3 on page 339 and Table 13-4 on page 339. The second disposition schedule requires you to add an alternative retention schedule. Refer to 13.2.4, “Adding an alternate retention schedule” on page 339.

Table 13-3 Contract Expired + 3 Y disposition schedule

Field information	**Value**
Schedule Name	Contract Expired + 3Y
Description	Destroy 3 years after contact is expired.
Internal Event	Contract Expired
Phase (1)	Destroy 3 years after the contract is closed: Phase Name: Destroy Phase Action: Destroy Action Is Screening Required: False Default Retention: 3 Years 0 Months 0 Days
CutOff Base	XYZContractExpirationDate

Table 13-4 Claim Closed + 5Y (ALT) disposition schedule

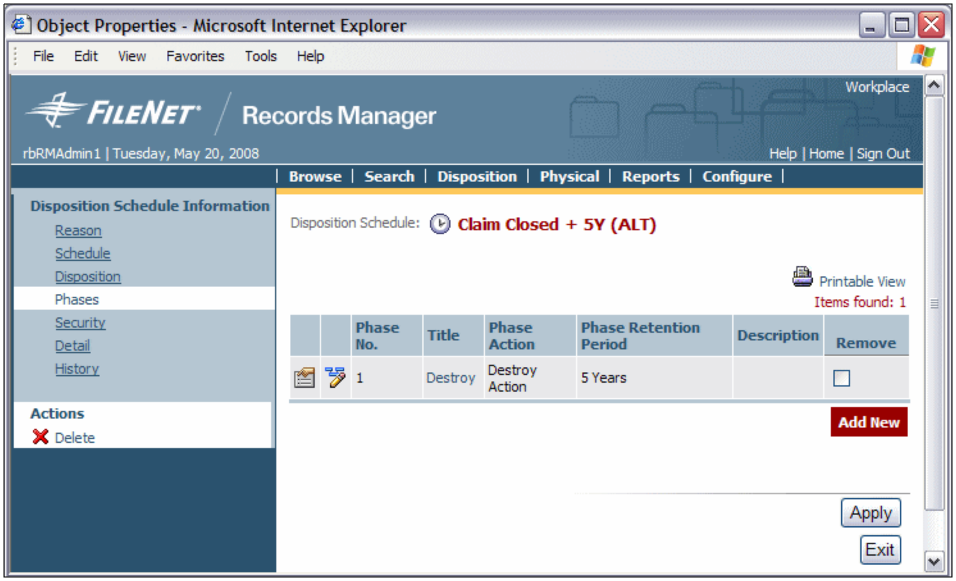
Field information	**Value**
Schedule Name	Claim Closed + 5Y (ALT)
Internal Event	Claim Closed
Phase (1)	Destroy 5 years after the insurance claim is closed (default): Phase Name: Destroy Phase Action: Destroy Action Is Screening Required: False Default Retention: 5 Years 0 Months 0 Days
CutOff Base	XYZClaimCloseDate
Alternate retention	Based on XYZState (Record folder property) where If XYZState=CA, then keep for 7 years after the claim is closed. If XYZState=FL, then keep for 3 years after the claim is closed.

13.2.4 Adding an alternate retention schedule

In our case study, the disposition schedule Claim Closed + 5Y (ALT) has alternate retentions. In general, we destroy claim files 5 years after the insurance claim is closed. Alternatively, if the claim state is California, we must not destroy the claim files until 7 years after the claim is closed. If the claim state is Florida, the claim files only need to be kept for 3 years after the claim is closed.

To add alternate retention schedules, follow these steps:

1. Start from the disposition schedule that you created or that you are currently creating and click **Phases** from the left pane to go to the Phase page. Refer to Figure 13-20.


Object Properties - Microsoft Internet Explorer
File Edit View Favorites Tools Help
FILENET Records Manager
Workplace
rbRMAdmin1 | Tuesday, May 20, 2008
Help | Home | Sign Out
Browse | Search | Disposition | Physical | Reports | Configure
Disposition Schedule Information
Reason
Schedule
Disposition
Phases
Security
Detail
History
Actions
Delete
Printable View
Add New
Apply
Exit

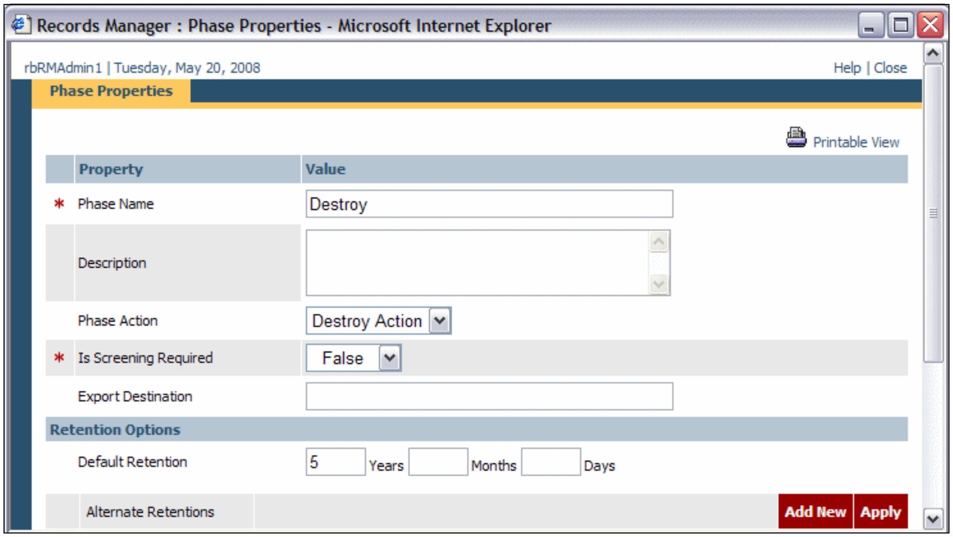
Disposition Schedule: Claim Closed + 5Y (ALT)

Items found: 1

		Phase No.	Title	Phase Action	Phase Retention Period	Description	Remove
		1	Destroy	Destroy Action	5 Years		□

Figure 13-20 Alternate retention periods: Initial page

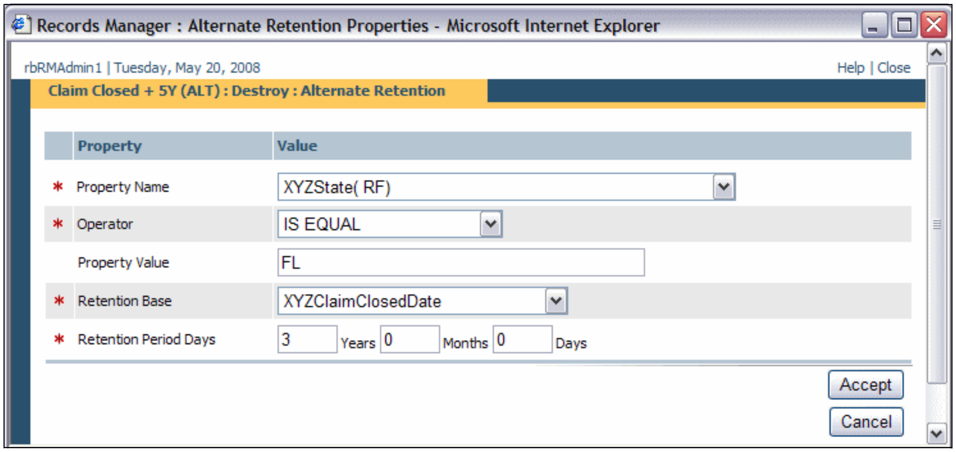
2. Click the existing phase link or click **Add New**. For our case study, we click the existing **Destroy** phase link. Figure 13-21 on page 341 shows the Phase Properties page.


Records Manager : Phase Properties - Microsoft Internet Explorer
rbRMAdmin1 | Tuesday, May 20, 2008
Help | Close
Phase Properties
Printable View
Add New
Apply

	Property	Value
*	Phase Name	Destroy
	Description	
	Phase Action	Destroy Action
*	Is Screening Required	False
	Export Destination	
Retention Options		
	Default Retention	5 Years Months Days
	Alternate Retentions	

Figure 13-21 Alternate retention periods: Initial phase properties page

3. Click **Add New** at the Alternate Retentions section.
4. Add the alternate retention period for the disposition schedule.
5. Select the property name, operator, and value for when the alternate retention period will be used. Also, select the retention base and retention periods. For our case study, we set XYZState to be equal to FL as the condition for the alternate retention period to be effective and 3 years as the new retention period. Refer to Figure 13-22 for the setup.

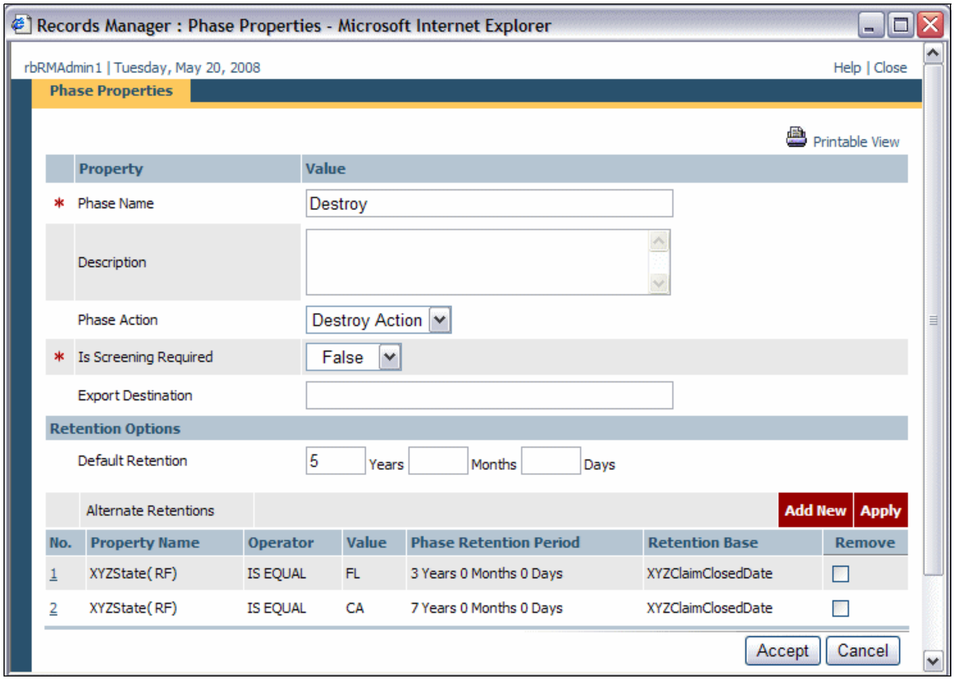

Records Manager : Alternate Retention Properties - Microsoft Internet Explorer
rbRMAdmin1 | Tuesday, May 20, 2008
Help | Close
Claim Closed + 5Y (ALT) : Destroy : Alternate Retention
Accept
Cancel

	Property	Value
*	Property Name	XYZState(RF)
*	Operator	IS EQUAL
	Property Value	FL
*	Retention Base	XYZClaimClosedDate
*	Retention Period Days	3 Years 0 Months 0 Days

Figure 13-22 Alternate retention periods: Set alternate condition and retention period

6. Click **Accept**.

   For our case study, we repeat the previous steps for the other alternate retention schedule when the state is California (Figure 13-23).



Records Manager : Phase Properties - Microsoft Internet Explorer

rbRMAdmin1 | Tuesday, May 20, 2008 Help | Close

Phase Properties

Printable View

	Property	Value
*	Phase Name	Destroy
	Description	
	Phase Action	Destroy Action
*	Is Screening Required	False
	Export Destination	

Retention Options

Default Retention: 5 Years, Months, Days

Alternate Retentions — Add New | Apply

No.	Property Name	Operator	Value	Phase Retention Period	Retention Base	Remove
1	XYZState(RF)	IS EQUAL	FL	3 Years 0 Months 0 Days	XYZClaimClosedDate	□
2	XYZState(RF)	IS EQUAL	CA	7 Years 0 Months 0 Days	XYZClaimClosedDate	□

Accept | Cancel

Figure 13-23 Alternate retention periods: Set other condition and retention periods

7. Click **Accept** again and click **Apply** to save the alternative retention periods for the disposition schedule.

13.3 Assigning the disposition schedule

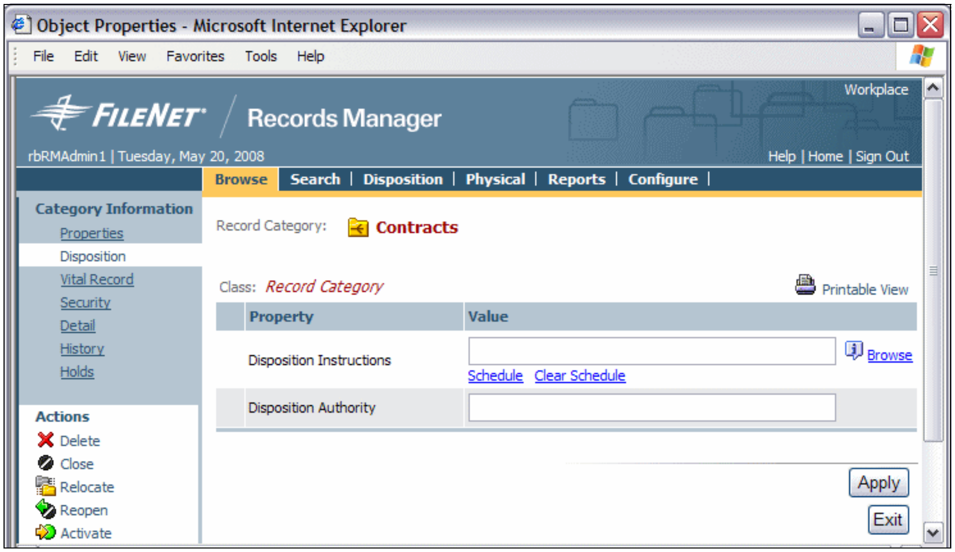
In order for the disposition schedule to be affective, you must assign it with a record category or record folder (either at the time of creation or later). Child containers inherit the disposition schedule from the parent containers.

To associate a disposition schedule later, perform the following steps:

1. Navigate to the file path container whose disposition schedule you want to assign and click the **Get Info** icon.

   For our case study, we navigate to the XYZ FilePlan/Finance/Contracts path.

2. Click **Disposition** in the left panel. Figure 13-24 shows the disposition schedule information for Contracts.



Object Properties - Microsoft Internet Explorer

File Edit View Favorites Tools Help

FILENET / Records Manager

Workplace

rbRMAdmin1 | Tuesday, May 20, 2008

Help | Home | Sign Out

Browse | Search | Disposition | Physical | Reports | Configure |

Category Information

- Properties
- Disposition
- Vital Record
- Security
- Detail
- History
- Holds

Actions

- Delete
- Close
- Relocate
- Reopen
- Activate

Record Category: Contracts

Class: *Record Category*

Printable View

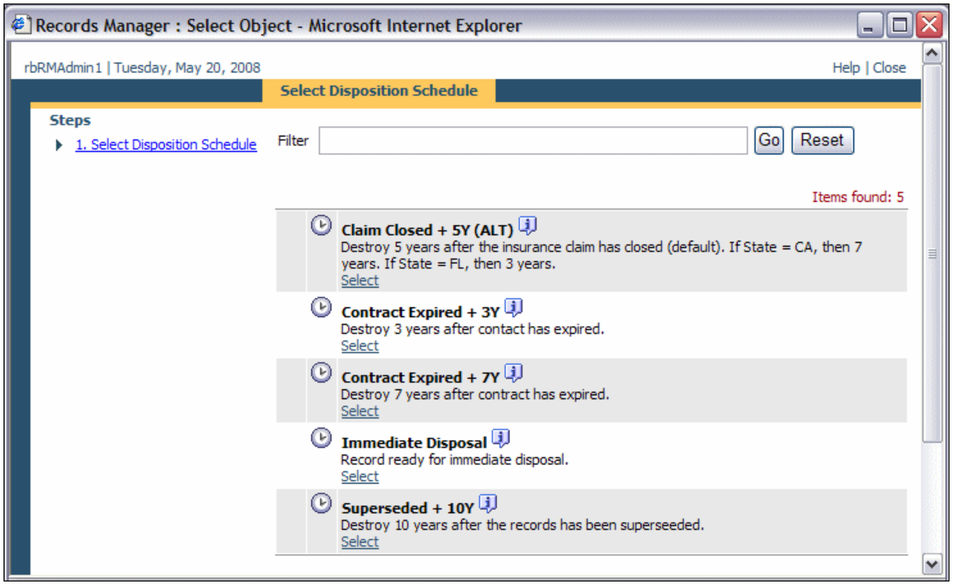
Property	Value
Disposition Instructions	Browse Schedule Clear Schedule
Disposition Authority	

Apply

Exit

Figure 13-24 Assign disposition schedule: Initial page

3. Click **Browse Schedule** to select a disposition schedule. Refer to Figure 13-25.



Records Manager : Select Object - Microsoft Internet Explorer

rbRMAdmin1 | Tuesday, May 20, 2008

Help | Close

Select Disposition Schedule

Steps

1. Select Disposition Schedule

Filter

Go Reset

Items found: 5

Claim Closed + 5Y (ALT)
Destroy 5 years after the insurance claim has closed (default). If State = CA, then 7 years. If State = FL, then 3 years.
Select

Contract Expired + 3Y
Destroy 3 years after contact has expired.
Select

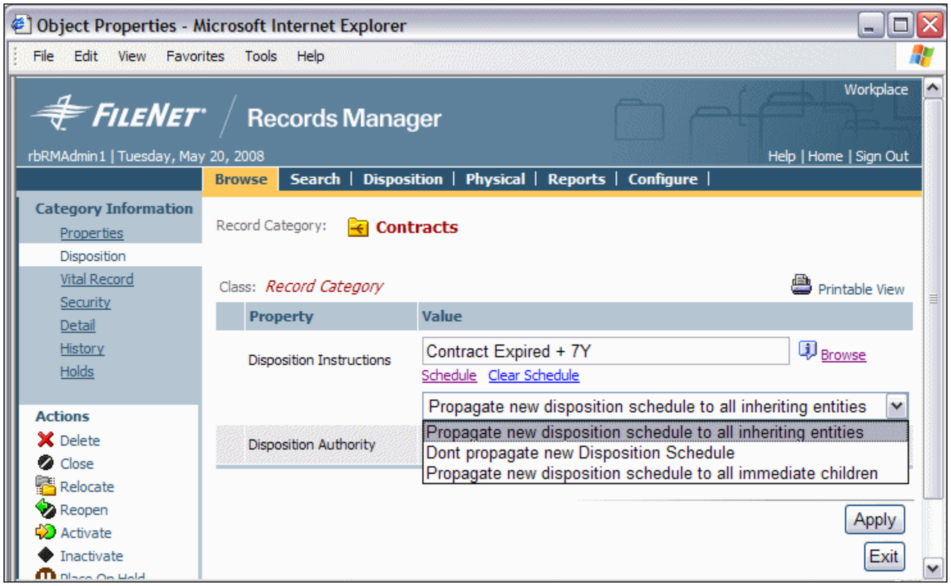
Contract Expired + 7Y
Destroy 7 years after contract has expired.
Select

Immediate Disposal
Record ready for immediate disposal.
Select

Superseded + 10Y
Destroy 10 years after the records has been superseeded.
Select

Figure 13-25 Assign disposition schedule: Select from a list of disposition schedules

4. Click **Select** under the name of a schedule. For our case study, we select **Contract Expired + 7Y** disposition schedule.
5. Select the appropriate option to indicate how you want to propagate the new disposition schedule to child containers. Refer to Figure 13-26.



Record Category: Contracts

Class: Record Category

Property	Value
Disposition Instructions	Contract Expired + 7Y Propagate new disposition schedule to all inheriting entities
Disposition Authority	

Propagate new disposition schedule to all inheriting entities
Dont propagate new Disposition Schedule
Propagate new disposition schedule to all immediate children

Figure 13-26 Assign disposition schedule: Propagation setup

When you assign a disposition schedule after creating the category or folder, you can propagate the schedule to its child containers in the following ways:

- Propagate new disposition schedule to all inheriting entities.

  This option propagates the schedule only to child containers that inherited the earlier disposition schedule. If the disposition schedule of child containers did not inherit the disposition schedule of the parent entity, these child containers’ disposition schedule remains unchanged.

- Propagate new disposition schedule to all immediate children.

  This option propagates the schedule to all child containers regardless of whether they inherited the earlier disposition schedule.

- Don’t propagate new disposition schedule.

  This option does not propagate the schedule to any of the child containers. If the child containers inherited the old disposition schedule, they will continue to be associated with that schedule but will not be categorized as inheriting entities.

For our case study, we select **Propagate new disposition schedule to all inheriting entities**.

6. Click **Apply** to save the modifications.
7. Click **Exit** to close the page.

13.3.1 Searching for categories assigned with disposition schedules

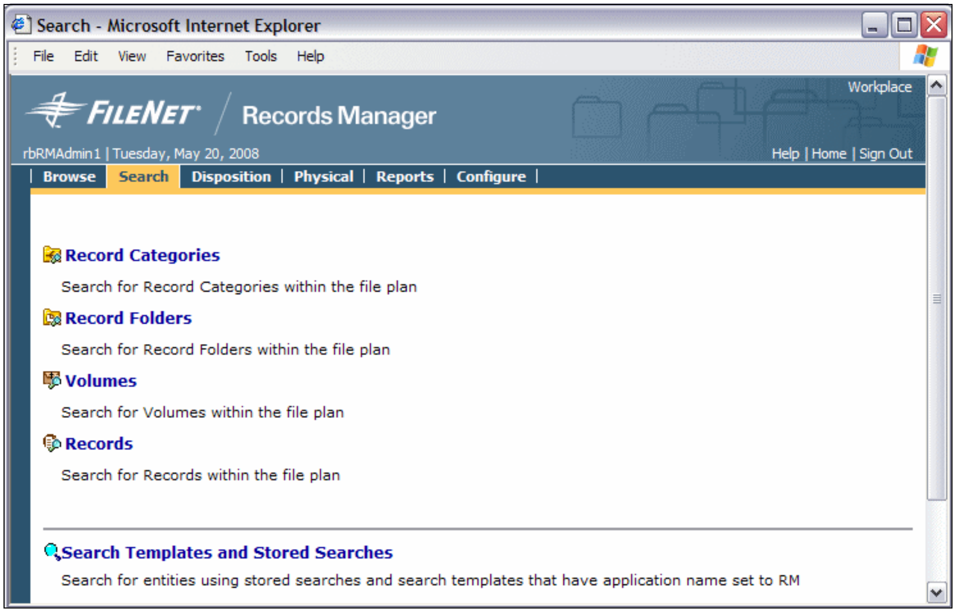
It is extremely important to be able to see what disposition schedules you have assigned to records entities. Table 13-5 shows the containers and their associated disposition schedules for our case study.

Table 13-5 Case study disposition schedules

Container	Disposition schedule	Type
XYZ FilePlan/Finance/ Contracts/	Contract Expired + 7Y	Associated
XYZ FilePlan/Finance/ Contracts/ Facility Management Contracts	Contract Expired + 3Y	Associated
XYZ FilePlan/Finance/ Contracts/ Service Contracts	Contract Expired + 7Y	Inherited from Contracts Record Category
XYZ FilePlan/Finance/ Contracts/ Supplier Contracts	Contract Expired + 7Y	Inherited from Contracts Record Category
XYZ FilePlan/Operations/ Claims	Claim Closed + 5Y (ALT)	Associated
XYZ FilePlan/Operations/ Claims/Auto	Claim Closed + 5Y (ALT)	Inherited from Claims Record Category
XYZ FilePlan/Operations/ Claims/Home	Claim Closed + 5Y (ALT)	Inherited from Claims Record Category
XYZ FilePlan/Operations/ Claims/Life	Claim Closed + 5Y (ALT)	Inherited from Claims Record Category

To search for disposition schedules assigned to entities, follow these steps:

1. Select the **Search** tab. Refer to Figure 13-27 on page 346.


Search - Microsoft Internet Explorer
File Edit View Favorites Tools Help
FILENET Records Manager
Workplace
rbRMAdmin1 | Tuesday, May 20, 2008
Help | Home | Sign Out
Browse Search Disposition Physical Reports Configure

Record Categories

Search for Record Categories within the file plan

Record Folders

Search for Record Folders within the file plan

Volumes

Search for Volumes within the file plan

Records

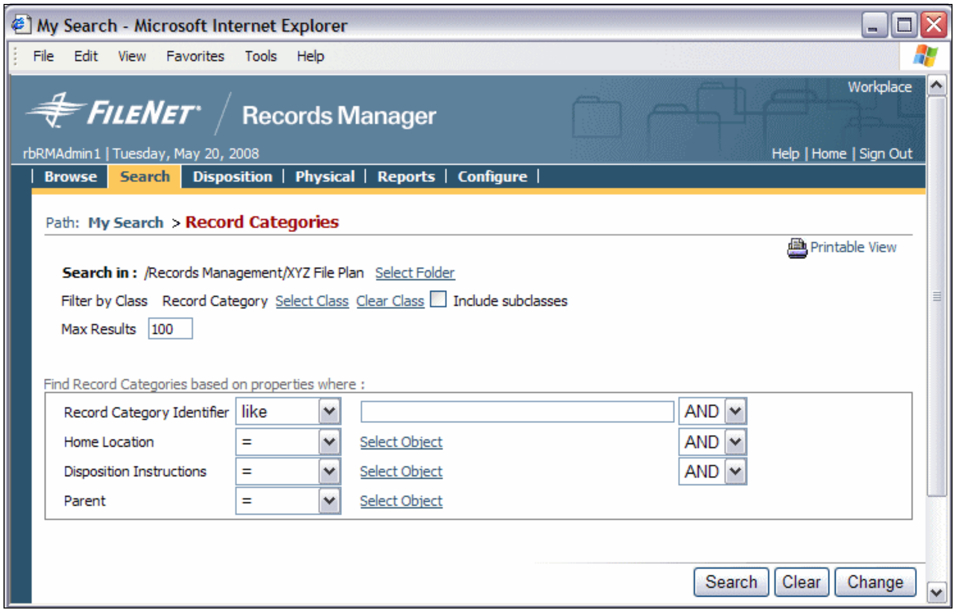
Search for Records within the file plan

Search Templates and Stored Searches

Search for entities using stored searches and search templates that have application name set to RM

Figure 13-27 Search entities assigned with disposition schedules

2. Click the entities where you want to perform search. For our case study, we select **Record Categories**.
3. Set the search criteria. Figure 13-28 on page 347 shows the default criteria window for searching against record categories.



Path: My Search > Record Categories

Search in : /Records Management/XYZ File Plan

Filter by Class Record Category ☐ Include subclasses

Max Results 100

Find Record Categories based on properties where :

Record Category Identifier	like		AND
Home Location	=	Select Object	AND
Disposition Instructions	=	Select Object	AND
Parent	=	Select Object	

Figure 13-28 Search entities assigned with disposition schedules: Set criteria

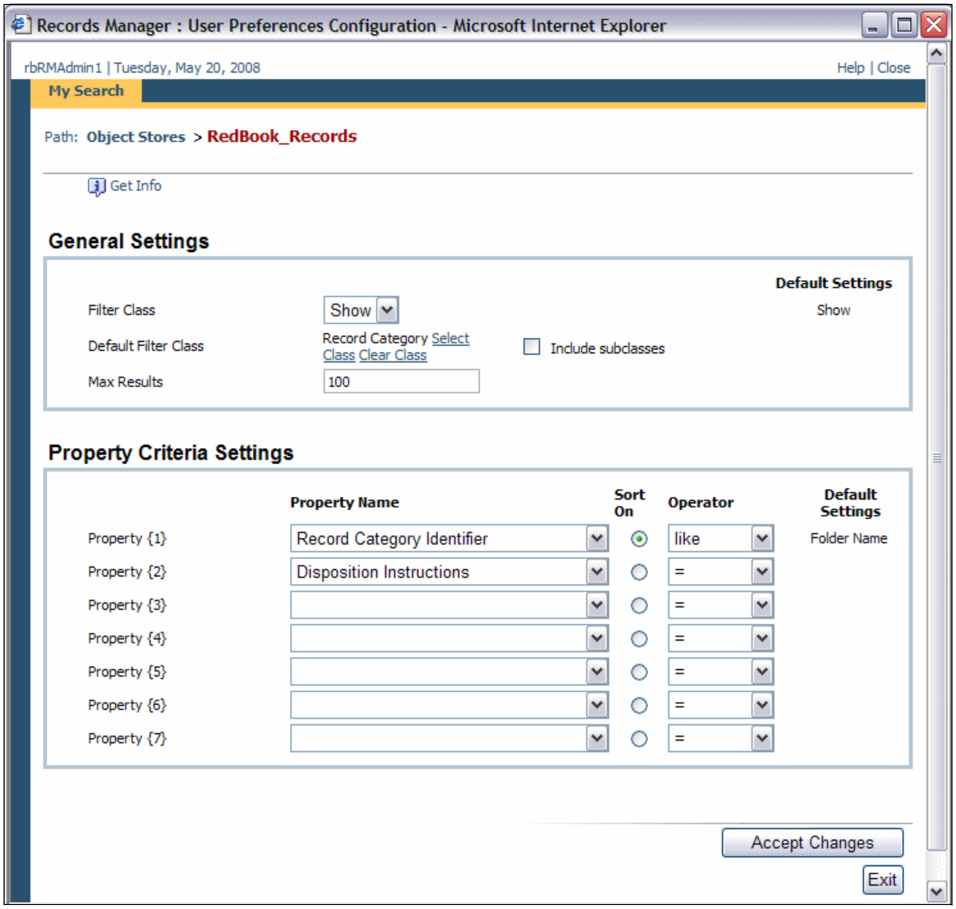
To customize the search properties for our case study:

a. Click **Change**.

b. Select only the properties that you need for your search criteria. For our case study, we keep the following properties for search criteria:

- Property {1} = Record Category Identifier
- Property {2} = Disposition Instructions

Refer to Figure 13-29 on page 348.


Records Manager : User Preferences Configuration - Microsoft Internet Explorer
rbRMAdmin1 | Tuesday, May 20, 2008
Help | Close
My Search
Get Info
Accept Changes
Exit

Path: Object Stores > RedBook_Records

General Settings

			Default Settings
Filter Class	Show		Show
Default Filter Class	Record Category Select Class Clear Class	□ Include subclasses	
Max Results	100		

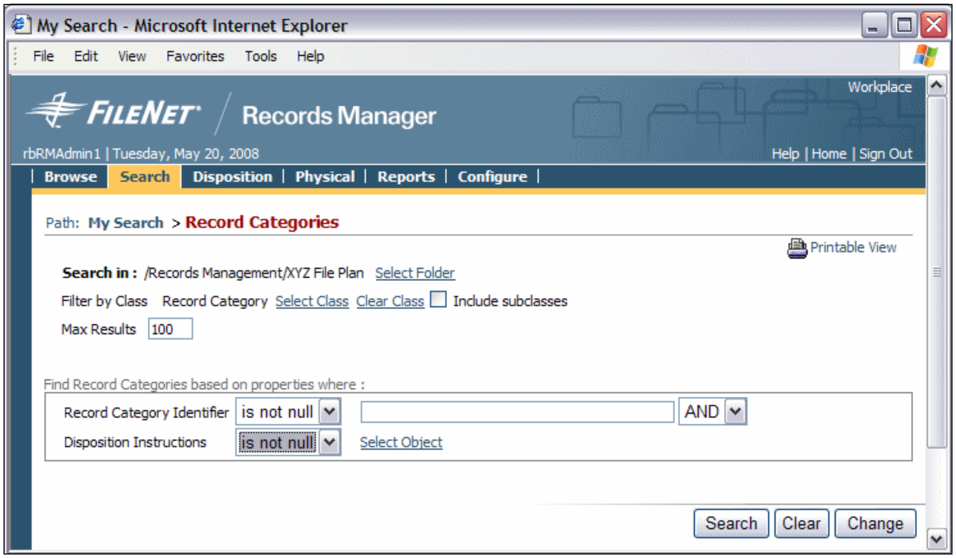
Property Criteria Settings

	Property Name	Sort On	Operator	Default Settings
Property {1}	Record Category Identifier	●	like	Folder Name
Property {2}	Disposition Instructions	○	=	
Property {3}		○	=	
Property {4}		○	=	
Property {5}		○	=	
Property {6}		○	=	
Property {7}		○	=	

Figure 13-29 Search entities assigned with disposition schedules: Change criteria

c. Click **Accept Changes**.

d. Set your criteria. For our case study, we select **Is Not Null** for both properties. Refer to Figure 13-30 on page 349.


My Search - Microsoft Internet Explorer
File Edit View Favorites Tools Help
FileNet Records Manager
Workplace
Help | Home | Sign Out
Browse | Search | Disposition | Physical | Reports | Configure
Printable View
Search Clear Change

rbRMAdmin1 | Tuesday, May 20, 2008

Path: My Search > **Record Categories**

Search in : /Records Management/XYZ File Plan Select Folder

Filter by Class Record Category Select Class Clear Class □ Include subclasses

Max Results 100

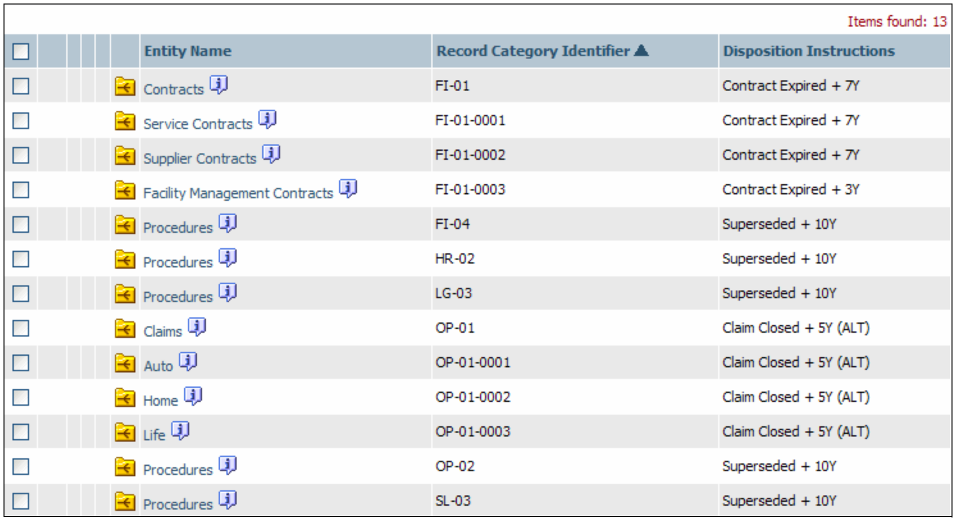
Find Record Categories based on properties where :

Record Category Identifier is not null AND

Disposition Instructions is not null Select Object

Figure 13-30 Search entities assigned with disposition schedules: Search for all

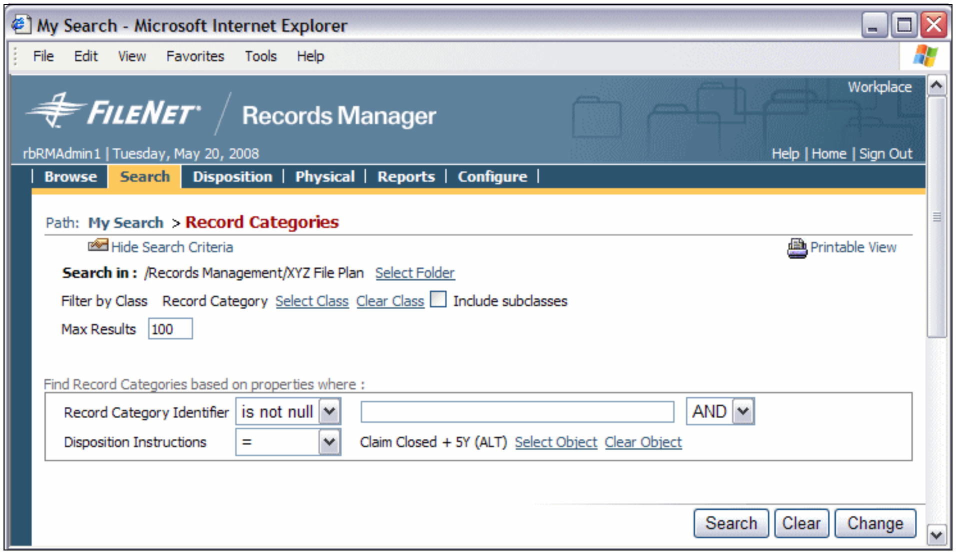
4. Click **Search**. Figure 13-31 shows the search result.


Items found: 13

□	Entity Name	Record Category Identifier ▲	Disposition Instructions
□	Contracts	FI-01	Contract Expired + 7Y
□	Service Contracts	FI-01-0001	Contract Expired + 7Y
□	Supplier Contracts	FI-01-0002	Contract Expired + 7Y
□	Facility Management Contracts	FI-01-0003	Contract Expired + 3Y
□	Procedures	FI-04	Superseded + 10Y
□	Procedures	HR-02	Superseded + 10Y
□	Procedures	LG-03	Superseded + 10Y
□	Claims	OP-01	Claim Closed + 5Y (ALT)
□	Auto	OP-01-0001	Claim Closed + 5Y (ALT)
□	Home	OP-01-0002	Claim Closed + 5Y (ALT)
□	Life	OP-01-0003	Claim Closed + 5Y (ALT)
□	Procedures	OP-02	Superseded + 10Y
□	Procedures	SL-03	Superseded + 10Y

Figure 13-31 Search entities assigned with disposition schedules: Results

If you want to search entities that have been assigned to a specific disposition schedule, for example, Claim Closed + 5Y (ALT), set up your search criteria as shown in Figure 13-32 on page 350.


My Search - Microsoft Internet Explorer
File Edit View Favorites Tools Help
FILENET Records Manager
Workplace
rbRMAdmin1 | Tuesday, May 20, 2008
Help | Home | Sign Out
Browse | Search | Disposition | Physical | Reports | Configure
Hide Search Criteria
Printable View
Select Folder
Select Class
Clear Class
Select Object
Clear Object
Search
Clear
Change

Path: My Search > Record Categories

Search in : /Records Management/XYZ File Plan

Filter by Class: Record Category ☐ Include subclasses

Max Results: 100

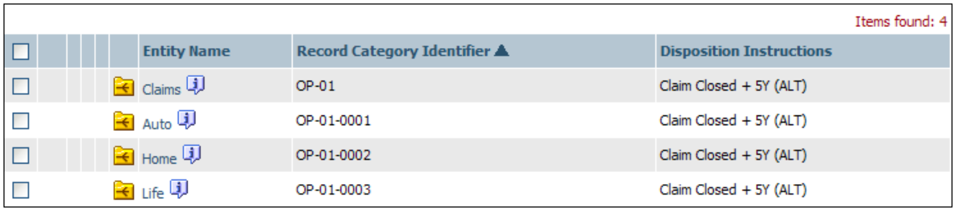
Find Record Categories based on properties where :

Record Category Identifier: is not null AND

Disposition Instructions: = Claim Closed + 5Y (ALT)

Figure 13-32 Search entities assigned with disposition schedules: Specific search

Figure 13-33 shows the result of the search.



Items found: 4

☐	Entity Name	Record Category Identifier ▲	Disposition Instructions
☐	Claims	OP-01	Claim Closed + 5Y (ALT)
☐	Auto	OP-01-0001	Claim Closed + 5Y (ALT)
☐	Home	OP-01-0002	Claim Closed + 5Y (ALT)
☐	Life	OP-01-0003	Claim Closed + 5Y (ALT)

Figure 13-33 Search entities assigned with disposition schedules: Result 2

13.4 Running the Disposition Sweep

After you assign a disposition schedule to file plan containers, you can run Disposition Sweep (periodically) to have the system check which records are ready to be moved through the disposition process.

Before you can run Disposition Sweep, you must configure it for the appropriate values.

13.4.1 Configuring the Disposition Sweep

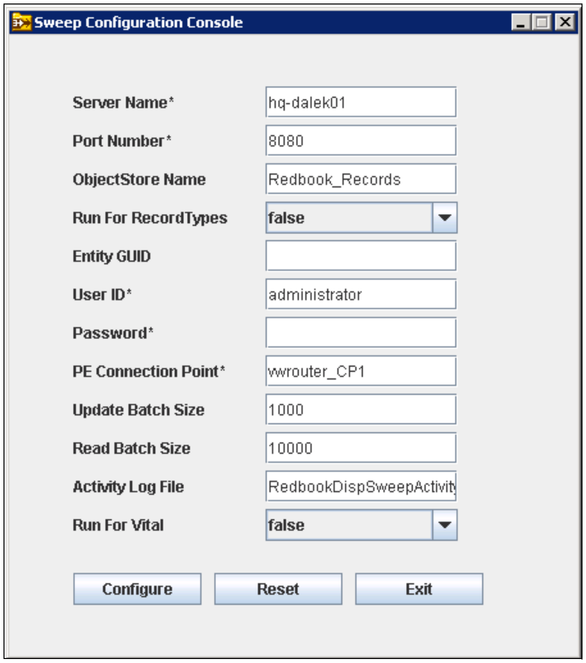
You can configure Disposition Sweep to run against a specific File Plan Object Store (FPOS) or all of them in a Content Engine (CE). To configure Disposition Sweep, perform the following tasks:

1. From a command prompt, navigate to the RecordsManagerSweep folder and execute the following command:

   ```
   RecordsManagerSweep.bat -DispositionSweep -configure
   ```

   For our case study, we create a separate directory for the HoldSweep folder. It is under C:\Program Files\FileNet\RM\RedbookRecordsManagerSweep>

   The Disposition Sweep Configuration Console dialog box appears. Refer to Figure 13-34.



Server Name*	hq-dalek01
Port Number*	8080
ObjectStore Name	Redbook_Records
Run For RecordTypes	false
Entity GUID	
User ID*	administrator
Password*	
PE Connection Point*	vwrouter_CP1
Update Batch Size	1000
Read Batch Size	10000
Activity Log File	RedbookDispSweepActivity
Run For Vital	false

Figure 13-34 Configuring Disposition Sweep

2. Specify the appropriate values for the following fields:
   - Server Name: Name or IP address of the Content Engine server.

     For our case study, it is hq-daleq01.
   - Port Number: The Web Services Interoperability (WSI) port number that is used by your Content Engine server.

The default port number for Content Engine running under WebSphere is 9080, under WebLogic is 7001, and under JBoss is 8080.

For our case study, we use 8080.

- ObjectStore Name (optional field): Globally Unique Identifier (GUID) or the name of the File Plan Object Store (FPOS) on which you want to run Disposition Sweep.

  If you do not provide a value, the Disposition Sweep process will run on *all* the File Plan Object Stores associated with the specified Content Engine server. If the name of the object store contains extended characters, use the GUID instead of the name.

  For our case study, it is RedBook_records.

  GUID is the Globally Unique Identifier (GUID) of the IBM FileNet P8 domain. Every Content Engine object has a GUID that cannot be changed.

- Run For Record Types:

  - Select True to have the Disposition Sweep process check all record types for any modifications made to the associated disposition schedules. If the disposition schedule of any record type has been modified, the Disposition Sweep process updates all the entities associated with that record type.
  - Select False to have the record types ignored. By default, record types are not processed.

  For our case study, we set this field to False.

- Entity GUID: GUID of the IBM FileNet Records Manager container for which you want to run the Disposition Sweep process. Disposition Sweep will run against the specified container and all of its children. By default, this node is empty, and all entities are processed.

  For our case study, we do not set any value in the field.

- User ID: User name that Disposition Sweep uses to log on to Content Engine to perform calculations.

  The user must have object store administrative rights on the FPOS and possess Records Administrator privileges.

  For our case study, we use rbRMAdmin1.

- Password: Password for the user ID.

- PE Connection Point: Name of the connection point that is created at the Content Engine server during installation.

  A *connection point* identifies a specific region of the workflow database, which contains both workflows and the data for all active workflows. This value is required for connection to the Process Engine server. (The person performing the installation creates this name.)

For our case study, we enter vwRouter_CP1.

- Update Batch Size: Number of entities to be updated as a batch using Disposition Sweep.

  By default, this value is set to 1000. For example, if this value is 1000 and there are 20,000 entities to be updated, Disposition Sweep will update all entities in 20 batches, with 1000 entities in each batch.

  For our case study, we use the default.

- Read Batch Size: Number of entities to be read per batch using Disposition Sweep.

  By default, this value is set to 100,000. For example, if this value is 100,000 and there are 1,000,000 entities to be read, all the entities will be read in 10 batches with 100,000 entities in each batch.

  For our case study, we use the default.

- Activity Log File: Name and path of the log file to be created by Disposition Sweep. By default, Disposition Sweep creates a file called DispositionSweepActivity.log in the ../FileNet/RecordsManagerSweep folder.

  For our case study, we use the default.

- Run For Vital:

  - Select True to check all record categories, records folders, volumes, and records for any modifications made to the associated vital metadata. If the disposition schedule of any entity has been modified, Disposition Sweep updates all the entities accordingly.

  - Select False to ignore the vital metadata. By default, vital metadata is not checked.

  For our case study, we use the default, False.

3. Click **Configure**. You see a message indicating the successful configuration of Disposition Sweep.

13.4.2 Executing the Disposition Sweep

Note: Before running Disposition Sweep, we manually set either the claim close date or the contract expiration date to 3/1/02 for every entity:

```
XYZClaimClosedDate=3/1/02
XYZContractExpirationDate=3/1/02
```

This way, Disposition Sweep can find entities that are ready for disposition.

Table 13-6 lists the entities, their entity values, and the associated disposition instructions. Cutoff dates and phase actions are not set for the record entities.

Table 13-6 Before Disposition Sweep

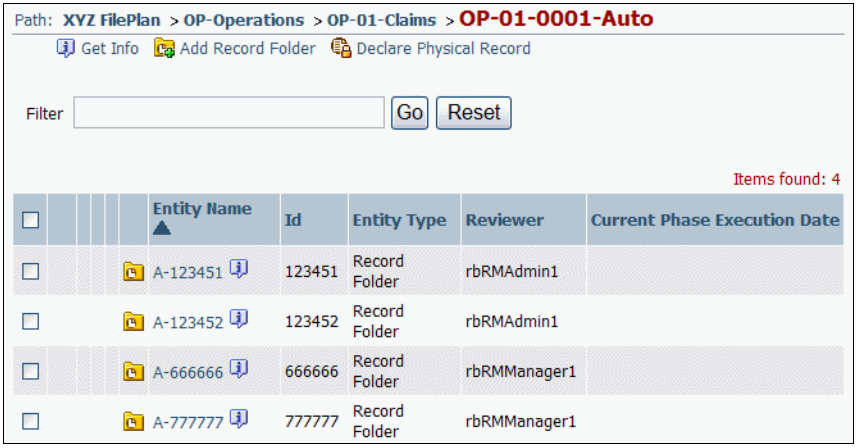
Entity type and name	Entity values	Disposition Instructions	Cutoff date	Current phase action

Record folder: A-666666/	XYZState=FL
	XYZClaimClosedDate =3/1/02
		Claim Closed + 5Y (ALT) Florida alternate 3 years	Not set	Not set

Record folder: A-777777/	XYZState=CA
	XYZClaimClosedDate =3/1/02
		Claim Closed + 5Y (ALT) California alternate 7 years	Not set	Not set

Record: Facility Management Contract 0001	XYZContractTypes= FacilityManagement XYZContractExpiration Date=3/1/02	Contract Expired + 3Y	Not set	Not set
Record: Service Contract 0001	XYZContractTypes= Services XYZContractExpiration Date=3/1/02	Contract Expired + 7Y	Not set	Not set

Notice that the entities in Table 13-6 currently do not have cutoff dates and do not have any phases assigned to them. Figure 13-35 on page 355, Figure 13-36 on page 355, and Figure 13-37 on page 355 show the record entities before the Disposition Sweep runs.

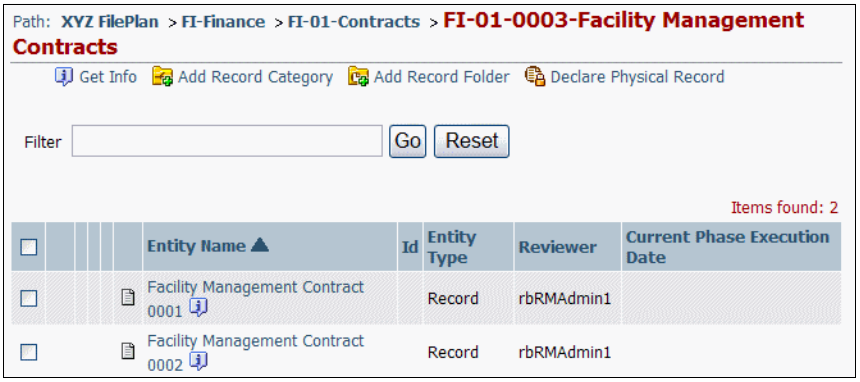


Path: XYZ FilePlan > OP-Operations > OP-01-Claims > OP-01-0001-Auto

Items found: 4

□	Entity Name ▲	Id	Entity Type	Reviewer	Current Phase Execution Date
□	A-123451	123451	Record Folder	rbRMAdmin1	
□	A-123452	123452	Record Folder	rbRMAdmin1	
□	A-666666	666666	Record Folder	rbRMManager1	
□	A-777777	777777	Record Folder	rbRMManager1	

Figure 13-35 Before Disposition Sweep run: Claim close + 5Y (ALT)

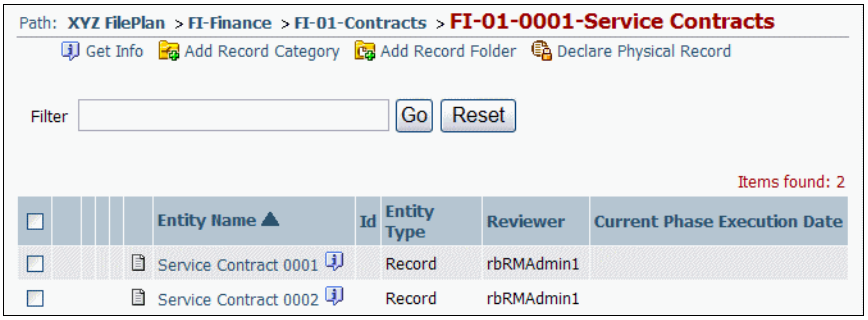


Path: XYZ FilePlan > FI-Finance > FI-01-Contracts > FI-01-0003-Facility Management Contracts

Items found: 2

□	Entity Name ▲	Id	Entity Type	Reviewer	Current Phase Execution Date
□	Facility Management Contract 0001		Record	rbRMAdmin1	
□	Facility Management Contract 0002		Record	rbRMAdmin1	

Figure 13-36 Before Disposition Sweep run: Contract expired + 3Y



Path: XYZ FilePlan > FI-Finance > FI-01-Contracts > FI-01-0001-Service Contracts

Items found: 2

□	Entity Name ▲	Id	Entity Type	Reviewer	Current Phase Execution Date
□	Service Contract 0001		Record	rbRMAdmin1	
□	Service Contract 0002		Record	rbRMAdmin1	

Figure 13-37 Before Disposition Sweep run: Contract expired + 7Y

To execute Disposition Sweep, follow these steps:

1. From a command prompt, navigate to the RecordsManagerSweep folder and enter the following command. In our case:

   ```
   C:\Program Files\FileNet\RM\RedbookRecordsManagerSweep>
   RecordsManagerSweep.bat -DispositionSweep
   ```

Table 13-7 shows the calculated cutoff date, current phase execution date, and phase action after running Disposition Sweep.

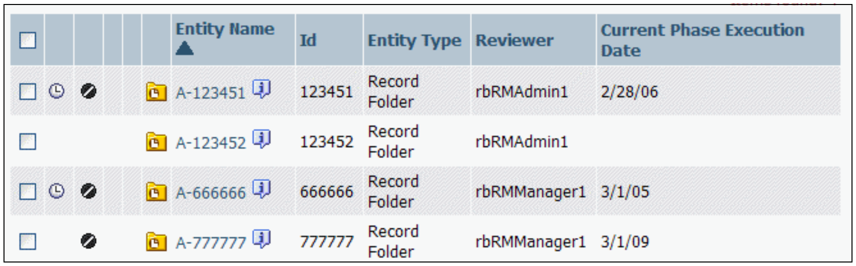
Table 13-7 After Disposition Sweep

Entity type and name	Entity values	Disposition instructions	Cutoff date	Current phase execution date	Current phase action
Record folder: A-666666	XYZState=FL XYZClaimClosedDate=3/1/02	Claim Closed + 5Y (ALT) FL alt 3 yrs	3/1/02	3/1/05	Destroy Action
Record folder: A-777777	XYZState=CA XYZClaimClosedDate=3/1/02	Claim Closed + 5Y (ALT) CA alt 7 yrs	3/1/02	3/1/09	Destroy Action
Record: Facility Management Contract 0001	XYZContractTypes= FacilityManagement XYZContractExpirationDate =3/1/02	Contract Expired + 3Y	3/1/02	3/1/05	Destroy Action
Record: Service Contract 0001	XYZContractTypes=Services XYZContractExpirationDate =3/1/02	Contract Expired + 7Y	Not set.[a]	3/1/09	Not set

[a] The cutoff date is not set. Disposition event offset is 7 years.

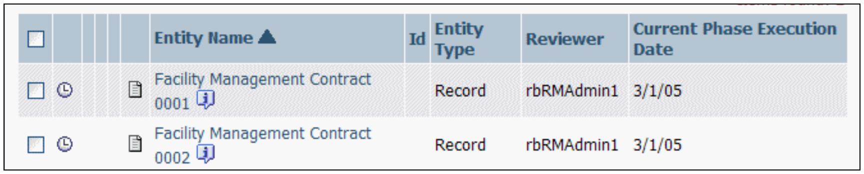
Figure 13-38 on page 357, Figure 13-39 on page 357, and Figure 13-40 on page 357 show our record entities after we run Disposition Sweep.

Notice that both folders A-666666 and A-777777 are closed. There is a Closed icon next to the folder. The first folder is also ready for disposition. There is a Clock icon next to the folder.



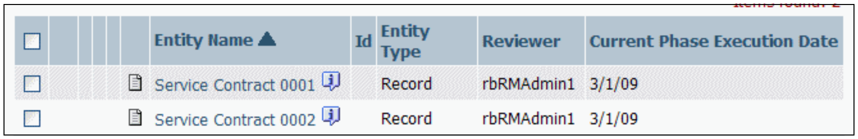
□			Entity Name ▲	Id	Entity Type	Reviewer	Current Phase Execution Date
□	◷	⊘	A-123451	123451	Record Folder	rbRMAdmin1	2/28/06
□			A-123452	123452	Record Folder	rbRMAdmin1	
□	◷	⊘	A-666666	666666	Record Folder	rbRMManager1	3/1/05
□		⊘	A-777777	777777	Record Folder	rbRMManager1	3/1/09

Figure 13-38 After Disposition Sweep run: Claim close + 5Y (ALT)



□		Entity Name ▲	Id	Entity Type	Reviewer	Current Phase Execution Date
□	◷	Facility Management Contract 0001		Record	rbRMAdmin1	3/1/05
□	◷	Facility Management Contract 0002		Record	rbRMAdmin1	3/1/05

Figure 13-39 After Disposition Sweep run: Contract expired + 3Y



□	Entity Name ▲	Id	Entity Type	Reviewer	Current Phase Execution Date
□	Service Contract 0001		Record	rbRMAdmin1	3/1/09
□	Service Contract 0002		Record	rbRMAdmin1	3/1/09

Figure 13-40 After Disposition Sweep run: Contract expired + 7Y

In summary, when Disposition Sweep ran, it found that all our case study entities were ready for disposition, because the fields XYZContractExpirationDate and XYZClaimClosedDate are not null. As a result, it starts moving entities through the various phases according to their disposition schedules.

13.4.3 Initiating a disposition on an item eligible for disposition

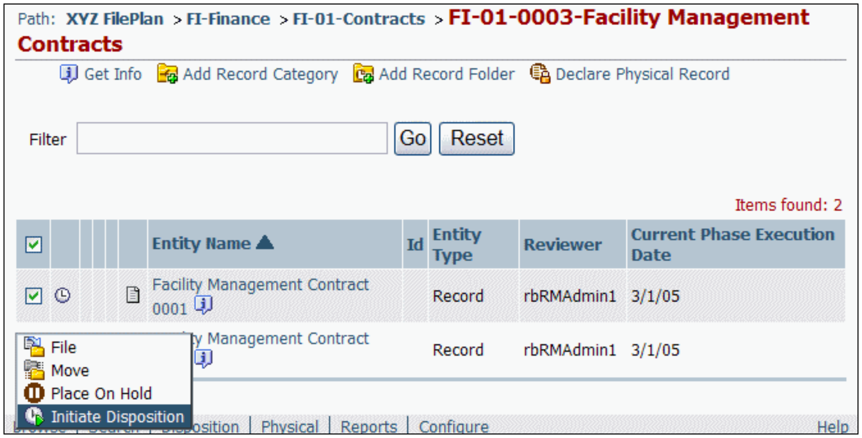
After the retention period is over, you must manually initiate the disposition action associated with that phase.

Initiating disposition on entities

To initiate a disposition schedule (refer to Figure 13-41 on page 358), follow these instructions:

1. Browse or search for entities that are ready for disposition. The Ready for Disposition icon appears beside entities that are ready.

2. Right-click the icon and select **Initiate Disposition** from the context menu. We select two entities. The system will group entities into batches, creating one workflow with both entities attached.



Path: XYZ FilePlan > FI-Finance > FI-01-Contracts > FI-01-0003-Facility Management Contracts

Get Info | Add Record Category | Add Record Folder | Declare Physical Record

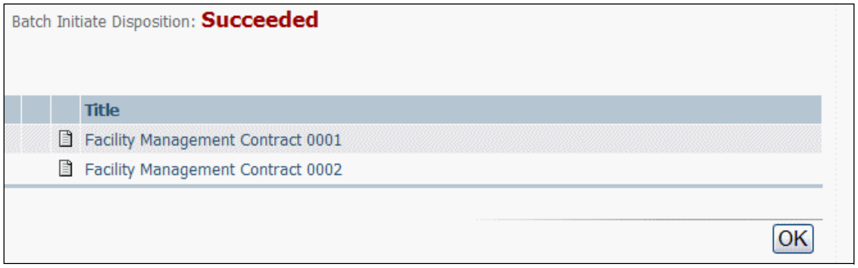
Filter [ ] Go Reset

Items found: 2

☑		Entity Name ▲	Id	Entity Type	Reviewer	Current Phase Execution Date
☑	◷	Facility Management Contract 0001		Record	rbRMAdmin1	3/1/05
		ty Management Contract		Record	rbRMAdmin1	3/1/05

Figure 13-41 Initiate Disposition

3. Click **OK** on the succeeded information window. Refer to Figure 13-42.

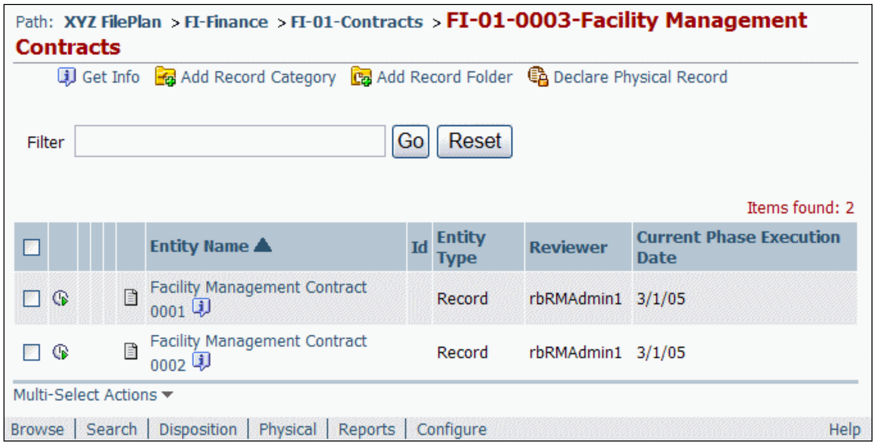


Batch Initiate Disposition: Succeeded

Title
Facility Management Contract 0001
Facility Management Contract 0002

Figure 13-42 Batch initiation

After we have initiated Disposition (refer to Figure 13-43 on page 359), the system launches the workflow associated with that disposition action and passes entities to the next phase on approval from an authorized person.



Path: XYZ FilePlan > FI-Finance > FI-01-Contracts > FI-01-0003-Facility Management Contracts

Items found: 2

□				Entity Name ▲	Id	Entity Type	Reviewer	Current Phase Execution Date
□				Facility Management Contract 0001		Record	rbRMAdmin1	3/1/05
□				Facility Management Contract 0002		Record	rbRMAdmin1	3/1/05

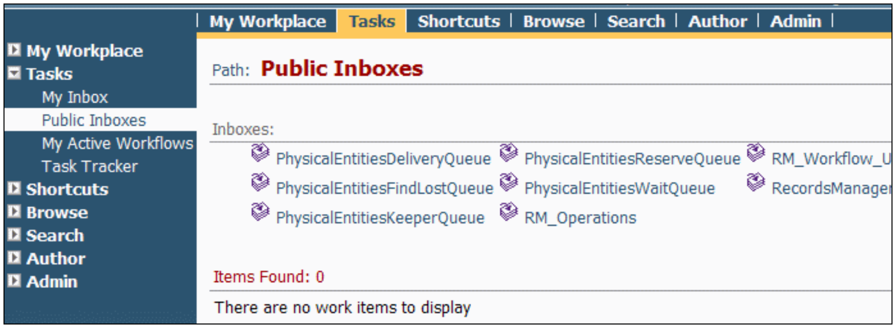
Figure 13-43 Disposition initiated

Processing items

Now, we can process items. In our case, that means destroy items using the disposition workflows:

1. Log on to Workplace with a user with access rights to the RecordsManagerApproval queue (refer to Figure 13-43).

2. Select **Tasks** from the Workplace tree navigator, and then, select **Public Inboxes** to access public work queues where work items can be selected (refer to Figure 13-44).

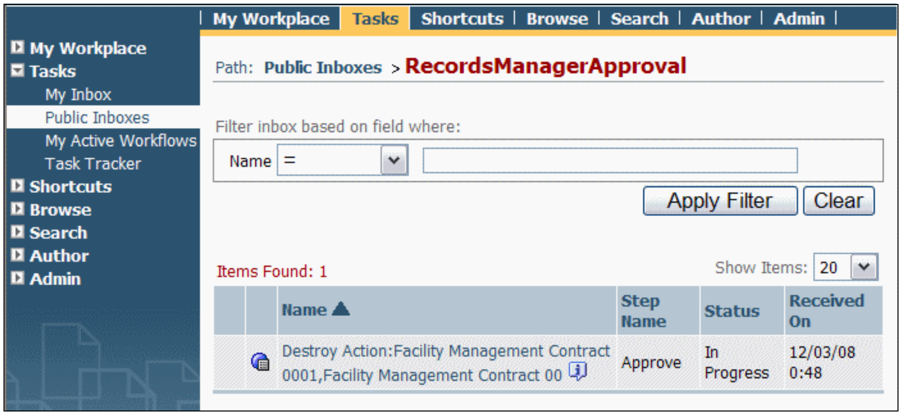

My Workplace | Tasks | Shortcuts | Browse | Search | Author | Admin
My Workplace
Tasks
My Inbox
Public Inboxes
My Active Workflows
Task Tracker
Shortcuts
Browse
Search
Author
Admin
Path: Public Inboxes
Inboxes:
PhysicalEntitiesDeliveryQueue
PhysicalEntitiesFindLostQueue
PhysicalEntitiesKeeperQueue
PhysicalEntitiesReserveQueue
PhysicalEntitiesWaitQueue
RM_Operations
RM_Workflow_U
RecordsManage

Items Found: 0

There are no work items to display

Figure 13-44 Tasks: Public Inboxes

3. Select the **RecordsManagerApproval** queue. Figure 13-45 on page 360 appears.



Path: Public Inboxes > RecordsManagerApproval

Filter inbox based on field where:

Name =

Items Found: 1

Show Items: 20

		Name ▲	Step Name	Status	Received On
		Destroy Action:Facility Management Contract 0001,Facility Management Contract 00	Approve	In Progress	12/03/08 0:48

Figure 13-45 RecordsManagerApproval public queue

4. Select the work item that was recently generated with a step name.
5. For our case study, we select **Destroy Action: Facility Management Contract 0001,Facility Management Contract 0002**.

   The step processor for that work item will appear.

   > **Note:** Step processors are designed to assist the user in completing the work step. There are many options related to general workflow step processing with Workplace. In this exercise, we focus on the features specifically related to IBM FileNet Records Manager.

   When the step processor appears, you see both records attached to the same workflow.

   The step processor shows the following instructions on the top of the window:

   ```
   This is the Review step of the Destroy Workflow.

   1. Click entity link to browse.

   2. Click Get Info link to review/modify the properties of the entity.

   3. Enter the Review Comments for approval/rejection of the entity.

   4. Select the Review Decision to approve/reject the destruction of
   the entity.
   ```

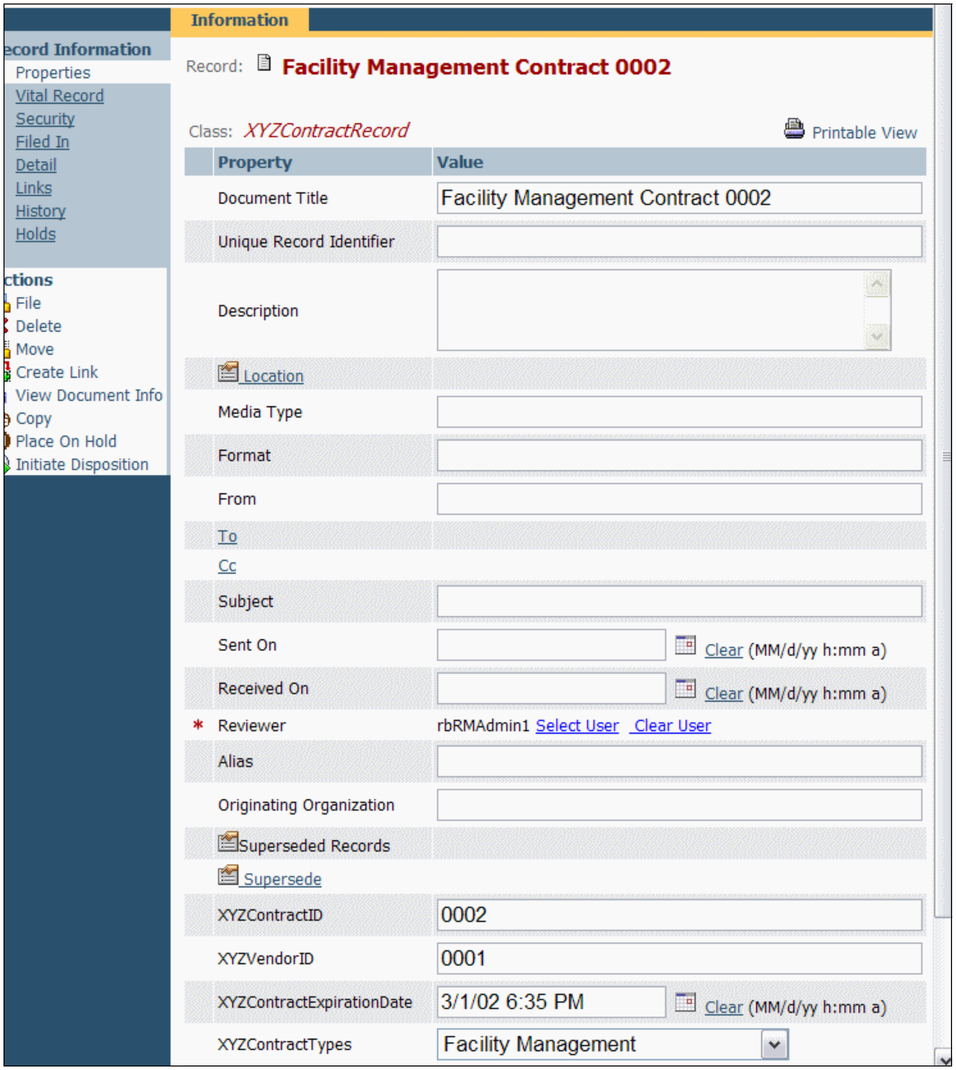
   Each entity has its own Review Comments and Review Decision (Approve, Reject, Hold, or Change Schedule). You can make separate decisions for each entity. The step processor also enables you to:

   – Reassign the work to another user by clicking **Reassign**.

   – Save your work/changes by clicking **Apply**.

- Complete or close the task by clicking **Complete** or **Close**.
- Check the current status by clicking **Status**. It shows all completed and incomplete workflow steps.

To view the record information, you can click **Get Info link**. Refer to Figure 13-46.



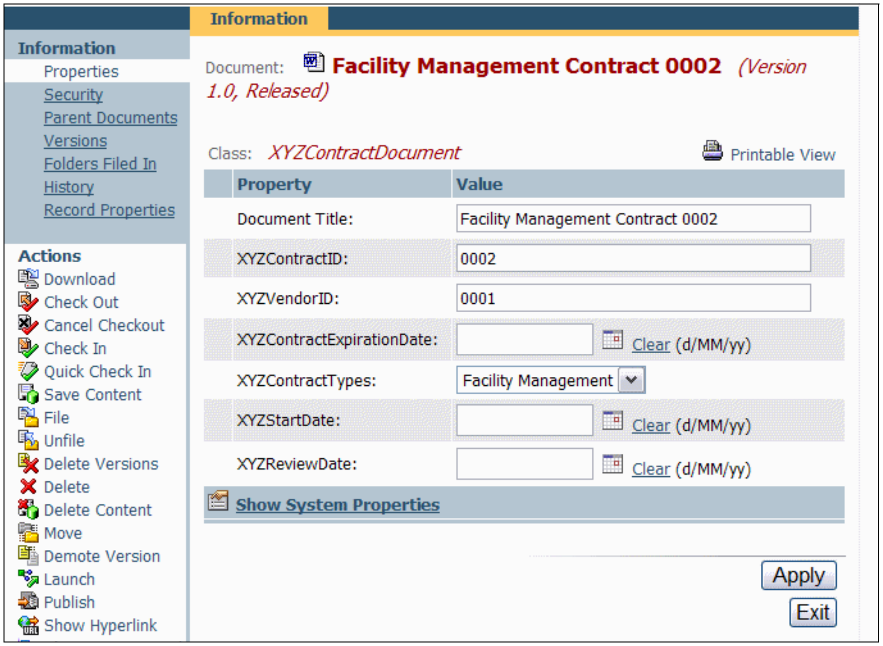
Record: Facility Management Contract 0002

Class: XYZContractRecord

Property	Value
Document Title	Facility Management Contract 0002
Unique Record Identifier	
Description	
Location	
Media Type	
Format	
From	
To	
Cc	
Subject	
Sent On	Clear (MM/d/yy h:mm a)
Received On	Clear (MM/d/yy h:mm a)
* Reviewer	rbRMAdmin1 Select User Clear User
Alias	
Originating Organization	
Superseded Records	
Supersede	
XYZContractID	0002
XYZVendorID	0001
XYZContractExpirationDate	3/1/02 6:35 PM Clear (MM/d/yy h:mm a)
XYZContractTypes	Facility Management

Figure 13-46 Record entity properties

From the Record Information page, we can click **View Document Info** to access to the properties of the related document that is stored in Content Engine. Refer to Figure 13-47.



Document: Facility Management Contract 0002 *(Version 1.0, Released)*

Class: *XYZContractDocument*

Property	Value
Document Title:	Facility Management Contract 0002
XYZContractID:	0002
XYZVendorID:	0001
XYZContractExpirationDate:	Clear (d/MM/yy)
XYZContractTypes:	Facility Management
XYZStartDate:	Clear (d/MM/yy)
XYZReviewDate:	Clear (d/MM/yy)

Figure 13-47 Document properties

6. Back to the workflow page, for Review Decision, select **Approve**, and click **Complete**.

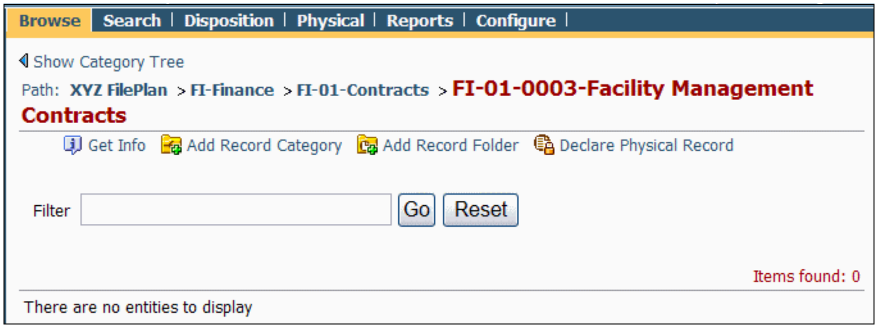
As a result, the item moves out of the RecordsManagerApproval queue. Now, the system automatically deletes both records from IBM FileNet Records Manager and its associated electronic document from Content Engine.

Verifying a destroy action

To ensure a proper destroy action, verify that the record entity is not in IBM FileNet Records Manager file repository any longer and that the associated documents are no longer in the Content Engine:

1. Verify that the record entities have been destroyed.

   Use IBM FileNet Records Manager to verify that the entities were successfully destroyed. From Workplace, browse to the **XYZ File Plan/Finance/Contracts/Facility Management Contracts** record category to verify that the two records that you initiated for disposition are no longer there. Refer to Figure 13-48 on page 363.



Path: XYZ FilePlan > FI-Finance > FI-01-Contracts > FI-01-0003-Facility Management Contracts

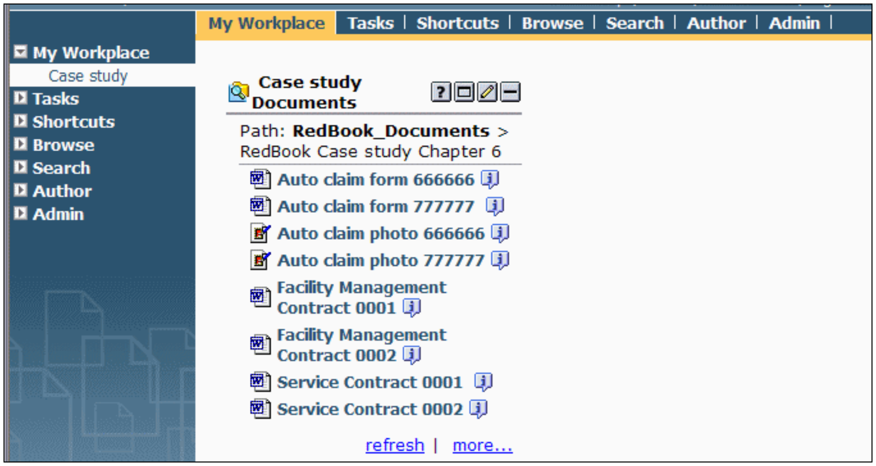
Items found: 0

There are no entities to display

Figure 13-48 Records were deleted

2. Verify that the associated documents are gone.

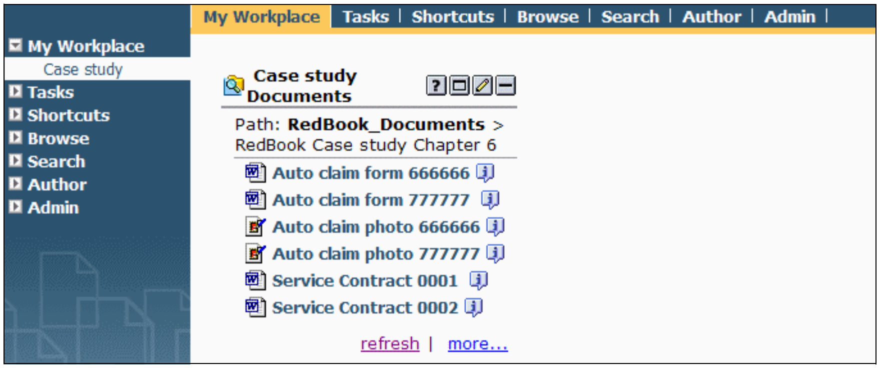
   Use Workplace to browse the **RedBook_Documents** → **RedBook Case study Chapter 6**. Verify that the two documents (Facility Management Contract 0001 and Facility Management Contract 0002) that you noted in Figure 13-43 on page 359 are no longer there. Refer to the records displayed in Workplace before and after the disposition execution (Figure 13-49 and Figure 13-50 on page 364).



Path: RedBook_Documents > RedBook Case study Chapter 6

Auto claim form 666666
Auto claim form 777777
Auto claim photo 666666
Auto claim photo 777777
Facility Management Contract 0001
Facility Management Contract 0002
Service Contract 0001
Service Contract 0002

Figure 13-49 Before approve destroy action



Path: **RedBook_Documents** > RedBook Case study Chapter 6

- Auto claim form 666666
- Auto claim form 777777
- Auto claim photo 666666
- Auto claim photo 777777
- Service Contract 0001
- Service Contract 0002

Figure 13-50 After approve destroy action

14

Records hold case study

In Chapter 7, “Records hold” on page 183, we discussed the concept of records hold. In this chapter, we show you the required steps to perform hold-related activities.

We describe the following topics in this chapter:

- Case study hold scenarios
- Creating a hold
- Placing and removing holds
- Hold Sweep:
  - Configuring Hold Sweep
  - Running Hold Sweep
  - Remove dynamic holds using Hold Sweep

14.1 Case study hold scenarios

Our case study is based on the fictitious Fictional Insurance Company X that we introduced in 3.3.2, “Example file plan” on page 65. The company’s file plan is shown in Figure 3-3 on page 66. The partial file plan showing categories that are related to the case study for this book is illustrated in Figure 14-1.

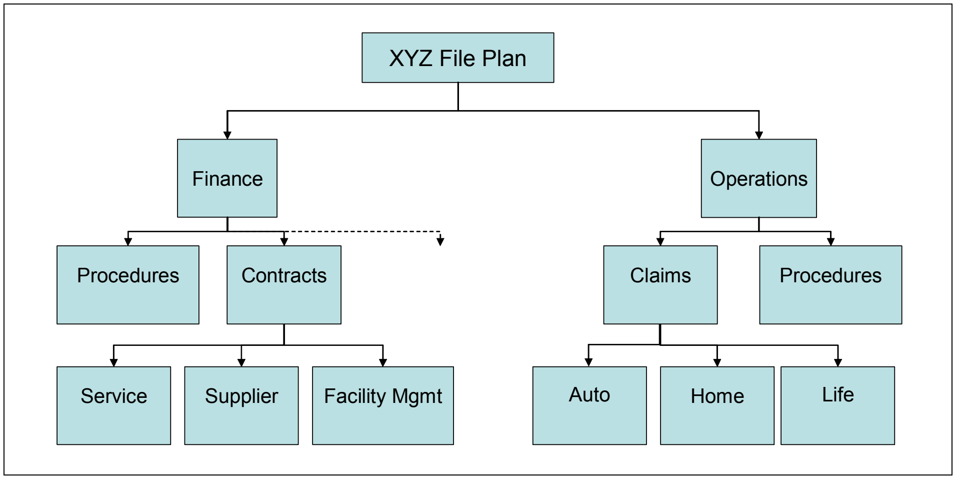


Figure 14-1 Partial file plan showing categories related to the case study for this book

Depending on litigation or business practices, your company might be asked to put specific records on hold. For our case study, we create several scenarios for when records might be put on hold:

- Lawsuit with contractors: In this scenario, there are lawsuits associated with contractors with whom the Fictional Insurance Company X works. The company is required to put all records associated with a vendor on hold.

  Case study hold condition: XYZVendor=0001

- Lawsuit with claims: In this scenario, there are disputes associated with particular claim cases. The company is required to put all records associated with a certain claim number on hold.

  Case study hold condition: XYZClaimNumber=666666

- Investigation of financial practices: In a general scenario, users might want to browse through the company’s file plan or perform searches on it and manually put several of the records on hold.

  Case study hold condition: None, because there is no predefined condition associated with this type of hold.

14.2 Creating a hold

In this section, we show you how to create a hold using one of the scenarios described in the previous section, the lawsuit with contractors. In this scenario, there are lawsuits associated with contractors with whom the Fictional Insurance Company X works. The company is required to put all records associated with a certain vendor on hold. The case study hold condition is XYZVendor=0001.

> **Note:** Always carefully consider the criteria that you use for dynamic holds so that a large number of entities are not placed on hold unintentionally.

To create a hold, follow these steps:

1. Launch IBM FileNet Records Manager Web application. Log on as a Records Manager or Records Administrator.
2. Select the **Disposition** tab. Refer to Figure 14-2.

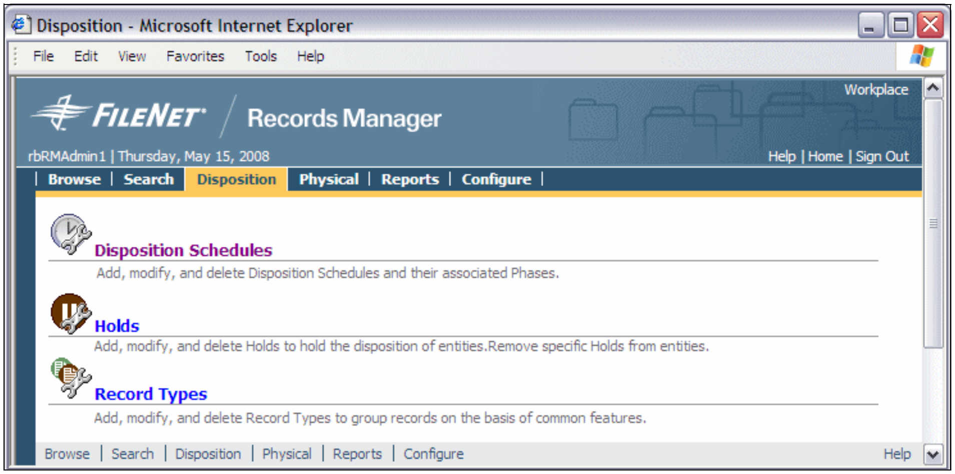

Disposition - Microsoft Internet Explorer
File Edit View Favorites Tools Help
FILENET Records Manager
Workplace
rbRMAdmin1 | Thursday, May 15, 2008
Help | Home | Sign Out
Browse | Search | Disposition | Physical | Reports | Configure
Disposition Schedules
Add, modify, and delete Disposition Schedules and their associated Phases.
Holds
Add, modify, and delete Holds to hold the disposition of entities.Remove specific Holds from entities.
Record Types
Add, modify, and delete Record Types to group records on the basis of common features.
Browse | Search | Disposition | Physical | Reports | Configure
Help

Figure 14-2 Disposition tab

3. Click **Holds**, and then, click **Add Hold**.
4. Set the hold properties:
   - Enter a name for the hold.

     For our case study, we enter Lawsuit with Contractors.
   - Specify a reason for adding this hold.

     For our case study, we enter RedBook case study.
   - Select the type of hold.

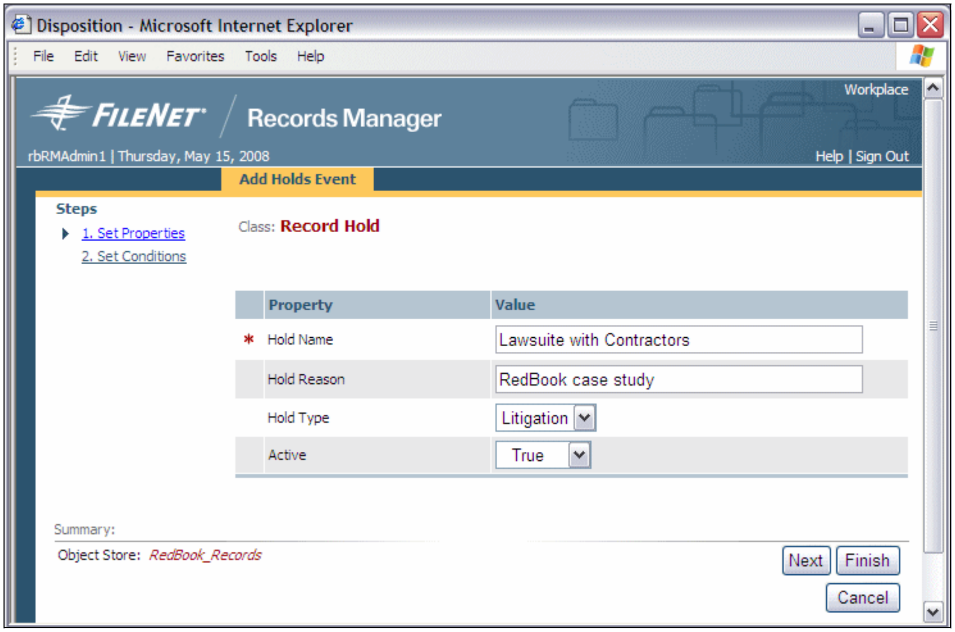
For our case study, we select **Litigation** from the drop-down list.

– From the Active drop-down menu, select **True**.

Selecting True enables the hold to be active immediately. Otherwise, the hold that you create is inactive and is not used by Hold Sweep during calculation.

For our case study, we set to True.

Figure 14-3 shows the setup for our case study.



Disposition - Microsoft Internet Explorer

rbRMAdmin1 | Thursday, May 15, 2008

Add Holds Event

Steps
1. Set Properties
2. Set Conditions

Class: Record Hold

	Property	Value
*	Hold Name	Lawsuite with Contractors
	Hold Reason	RedBook case study
	Hold Type	Litigation
	Active	True

Summary:
Object Store: RedBook_Records

Figure 14-3 Lawsuit with Contractors hold: Set Properties

5. Click **Next** to proceed to set the hold conditions.

   Alternatively, you can click **Finish** (or **Cancel**). If you finish without setting any conditions, this hold can only be placed manually. Hold Sweep will ignore it.

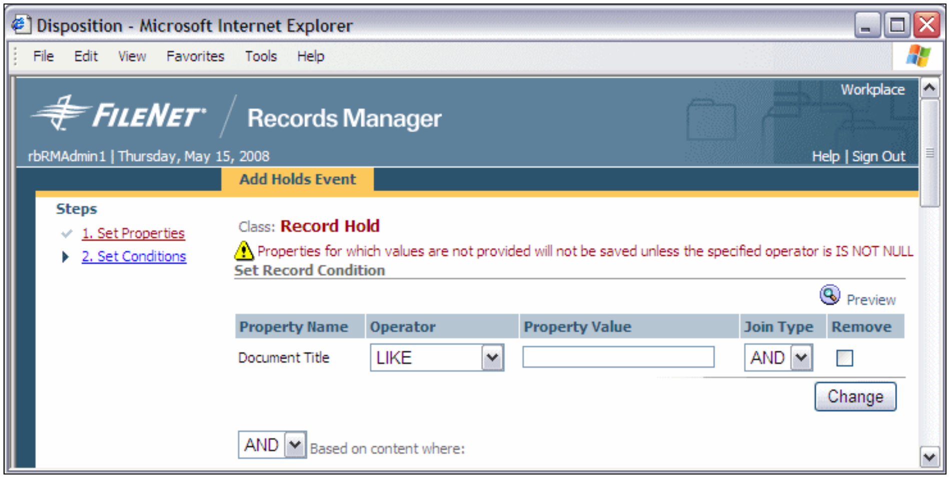
6. Set the hold conditions.

   You can set conditions for records, categories, records folders, or volumes. In each case, you specify one or more properties, an operator, a value, and a join type to specify the relationships between multiple properties. For records, you can also specify a content search. If you set conditions and the hold is active, running a Hold Sweep will automatically put entities that meet the criteria on hold.

To set the condition:

a. Click **Change** for the entity on which you want to specify conditions.

   For our case study, we want to change the condition to be based on XYZVendorID. We need to change the property name. Therefore, we click **Change**, which is associated with the Document Title. Refer to Figure 14-4.


Disposition - Microsoft Internet Explorer
File Edit View Favorites Tools Help
FILENET Records Manager
Workplace
rbRMAdmin1 | Thursday, May 15, 2008
Help | Sign Out
Add Holds Event
Steps
1. Set Properties
2. Set Conditions
Preview

Class: **Record Hold**

⚠ Properties for which values are not provided will not be saved unless the specified operator is IS NOT NULL

Set Record Condition

Property Name	Operator	Property Value	Join Type	Remove
Document Title	LIKE		AND	□

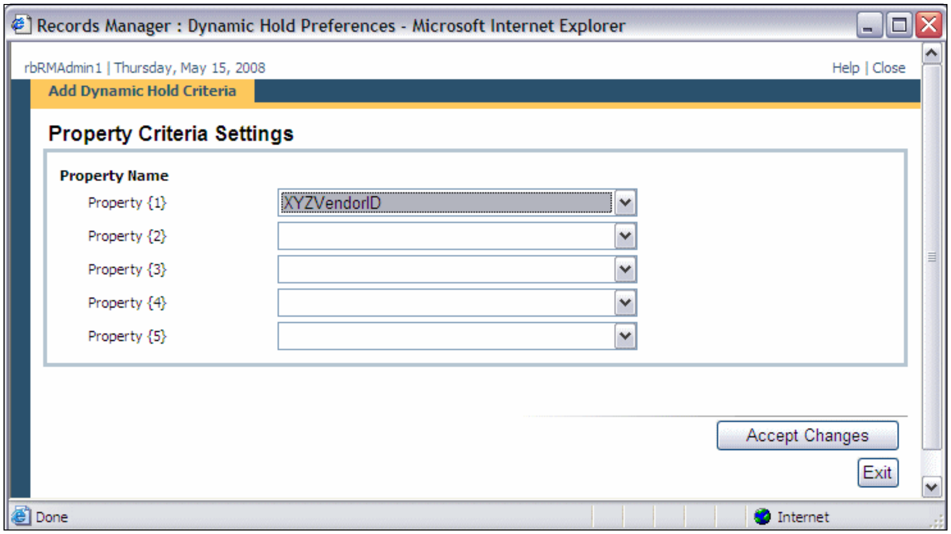
Change

AND Based on content where:

Figure 14-4 Lawsuit with Contractors hold: Set Conditions and change property name

b. From the drop-down menus, choose the property that you want to use.

   For our case study, we select **XYZVendorID**. Refer to Figure 14-5 on page 370.


Records Manager : Dynamic Hold Preferences - Microsoft Internet Explorer
rbRMAdmin1 | Thursday, May 15, 2008
Help | Close
Add Dynamic Hold Criteria
Accept Changes
Exit
Done
Internet

Property Criteria Settings

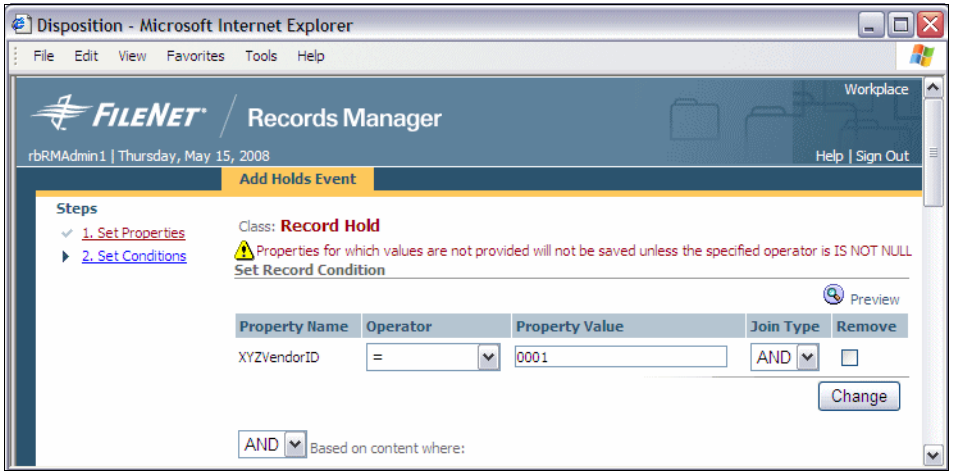
Property Name	
Property {1}	XYZVendorID
Property {2}	
Property {3}	
Property {4}	
Property {5}	

Figure 14-5 Lawsuit with Contractors hold: Set Conditions and select property name

c. Click **Accept Changes**.

d. Complete your condition by selecting the operator and the associated property value.

For our case study, our condition is to hold any records with XYZVendorID=0001. We set the operator to the equal sign and the property value to 0001. Refer to Figure 14-6.



Class: Record Hold

Properties for which values are not provided will not be saved unless the specified operator is IS NOT NULL

Set Record Condition

Property Name	Operator	Property Value	Join Type	Remove
XYZVendorID	=	0001	AND	□

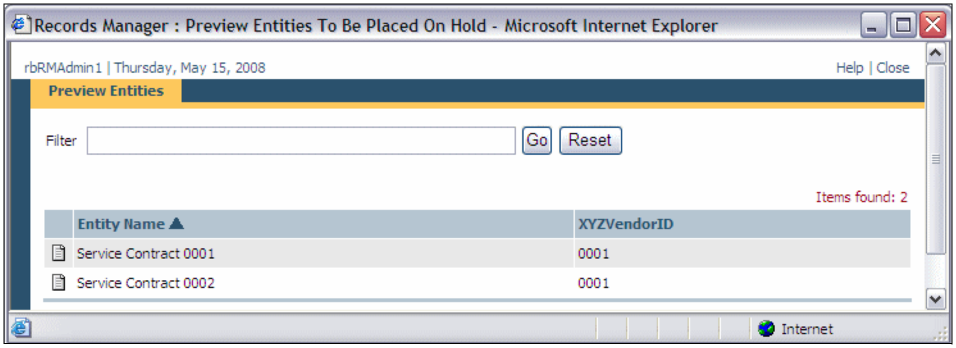
AND Based on content where:

Figure 14-6 Lawsuit with Contractors hold: Set Conditions XYZVendorID=0001

7. Click **Preview** to see previous entities that qualify for this condition.

   The system displays a list of entities that qualify for this condition. We recommend that you check the preview to make sure that you are placing holds on only the desired entities.

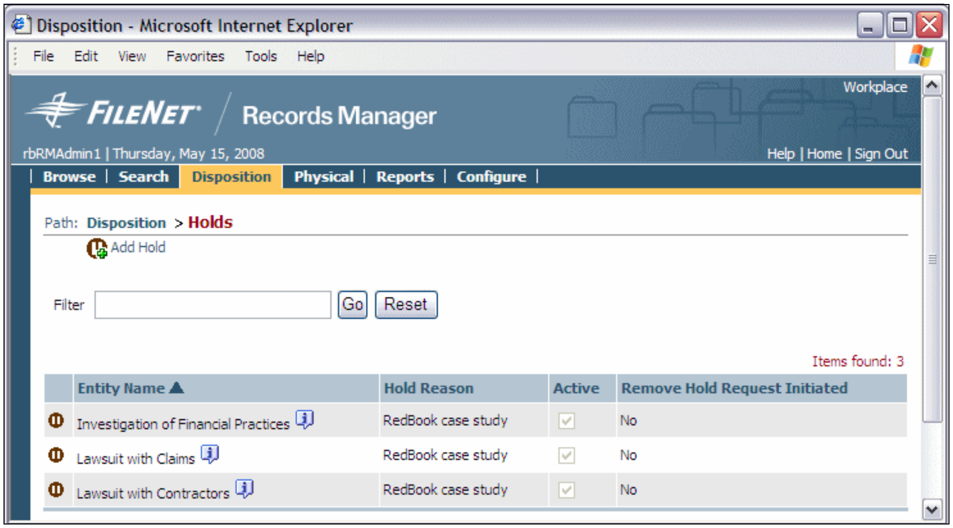
   For our case study, we have two contracts/records that belong to the 0001 Vendor ID. Refer to Figure 14-7.


Records Manager : Preview Entities To Be Placed On Hold - Microsoft Internet Explorer
rbRMAdmin1 | Thursday, May 15, 2008
Help | Close
Preview Entities
Filter
Go
Reset
Items found: 2
Internet

Entity Name ▲	XYZVendorID
Service Contract 0001	0001
Service Contract 0002	0001

Figure 14-7 Lawsuit with Contractors hold: Preview hold entities

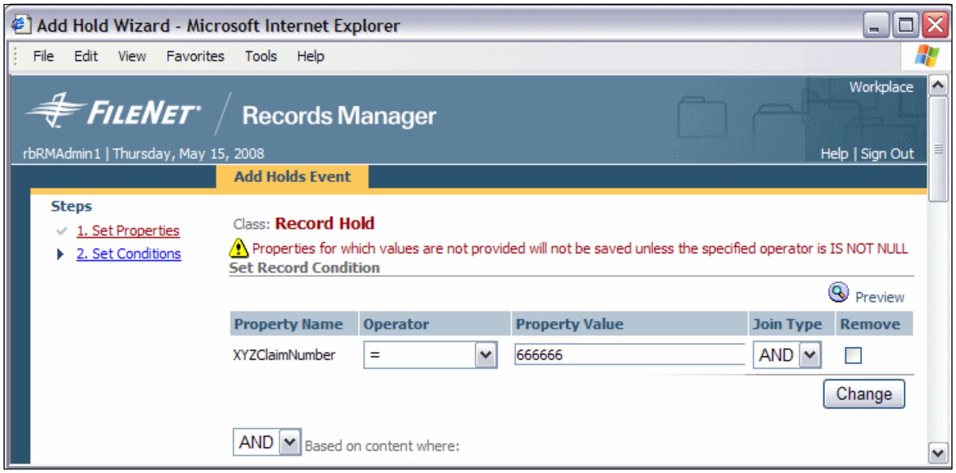
8. Click **Finish**. Figure 14-8 shows the list of holds for our case study.


Disposition - Microsoft Internet Explorer
File Edit View Favorites Tools Help
FILENET Records Manager
Workplace
rbRMAdmin1 | Thursday, May 15, 2008
Help | Home | Sign Out
Browse | Search | Disposition | Physical | Reports | Configure |
Path: Disposition > Holds
Add Hold
Filter
Go
Reset
Items found: 3

Entity Name ▲	Hold Reason	Active	Remove Hold Request Initiated
Investigation of Financial Practices	RedBook case study	☑	No
Lawsuit with Claims	RedBook case study	☑	No
Lawsuit with Contractors	RedBook case study	☑	No

Figure 14-8 Lawsuit with Contractors hold: Created

To create a hold for the lawsuit with Claims, use the condition XYZClaimNumber=666666. Refer to Figure 14-9 on page 372.


Add Hold Wizard - Microsoft Internet Explorer
File Edit View Favorites Tools Help
FileNet Records Manager
Workplace
rbRMAdmin1 | Thursday, May 15, 2008
Help | Sign Out
Add Holds Event
Steps
1. Set Properties
2. Set Conditions
Preview
Change

Class: **Record Hold**

⚠ Properties for which values are not provided will not be saved unless the specified operator is IS NOT NULL

Set Record Condition

Property Name	Operator	Property Value	Join Type	Remove
XYZClaimNumber	=	666666	AND	□

AND Based on content where:

Figure 14-9 Lawsuit with Claim hold: Set Conditions XYZClaimNumber=666666

14.3 Placing and removing holds

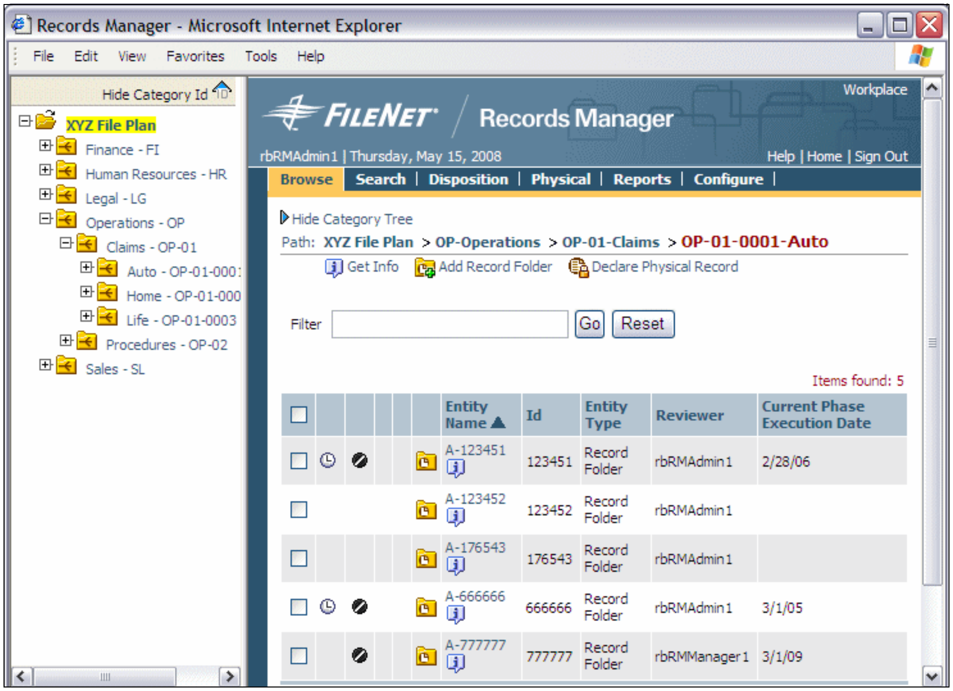
You can manually put entities on hold and manually remove the entities from being held. To place entities on hold automatically or *dynamically*, use Hold Sweep. Refer to 14.4, “Hold Sweep” on page 380.

14.3.1 Manually putting an entity on hold

To manually place a hold, follow these steps:

1. From IBM FileNet Records Manager Web application, navigate to the entities that you want to manually place on hold.

   For our case study, we have created several auto claims documents. We navigate to **XYZ File Plan** → **Operations** → **Claims** → **Auto**. Refer to Figure 14-10 on page 373.



Records Manager - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Hide Category Id

- XYZ File Plan
  - Finance - FI
  - Human Resources - HR
  - Legal - LG
  - Operations - OP
    - Claims - OP-01
      - Auto - OP-01-000:
      - Home - OP-01-000
      - Life - OP-01-0003
    - Procedures - OP-02
  - Sales - SL

FileNet Records Manager | Workplace

rbRMAdmin1 | Thursday, May 15, 2008 | Help | Home | Sign Out

Browse | Search | Disposition | Physical | Reports | Configure |

Hide Category Tree

Path: XYZ File Plan > OP-Operations > OP-01-Claims > OP-01-0001-Auto

Get Info | Add Record Folder | Declare Physical Record

Filter [ ] Go Reset

Items found: 5

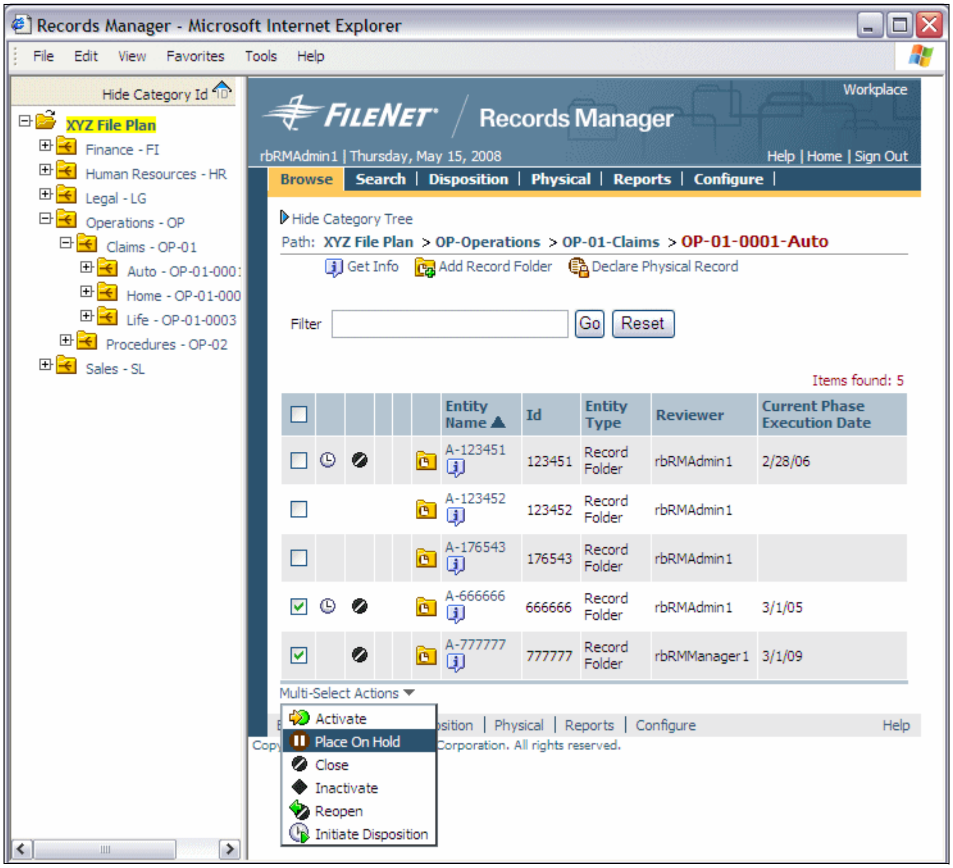
☐			Entity Name ▲	Id	Entity Type	Reviewer	Current Phase Execution Date
☐	◷	⊘	A-123451	123451	Record Folder	rbRMAdmin1	2/28/06
☐			A-123452	123452	Record Folder	rbRMAdmin1	
☐			A-176543	176543	Record Folder	rbRMAdmin1	
☐	◷	⊘	A-666666	666666	Record Folder	rbRMAdmin1	3/1/05
☐		⊘	A-777777	777777	Record Folder	rbRMManager1	3/1/09

Figure 14-10 Navigation

2. Select one or more entities that you want to put on hold.

   You can also right-click a single entity and use the context menu instead.

   For our case study, we put two items on hold: the records folders whose claim numbers equal A-666666 and A-777777. Refer to Figure 14-11 on page 374.

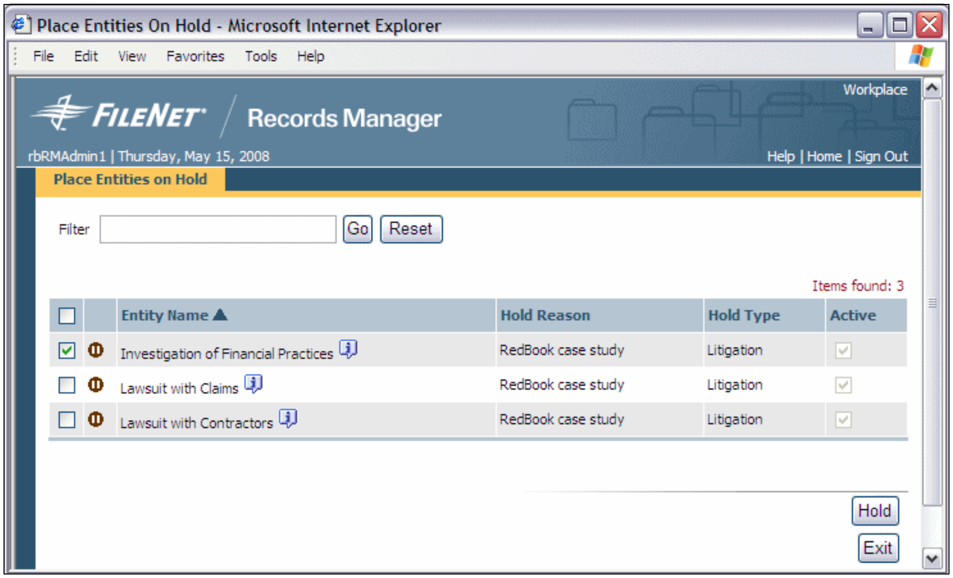


□			Entity Name ▲	Id	Entity Type	Reviewer	Current Phase Execution Date
□	◷	⊘	A-123451	123451	Record Folder	rbRMAdmin1	2/28/06
□			A-123452	123452	Record Folder	rbRMAdmin1	
□			A-176543	176543	Record Folder	rbRMAdmin1	
☑	◷	⊘	A-666666	666666	Record Folder	rbRMAdmin1	3/1/05
☑		⊘	A-777777	777777	Record Folder	rbRMManager1	3/1/09

Figure 14-11 Selecting entities to be placed on hold

3. From the Multi-Select Actions drop-down menu (or the context menu of the specific entity), select **Place On Hold**. Refer to Figure 14-11.
4. The Place Entities on Holds page appears (refer to Figure 14-12 on page 375). From the list of the available holds, select one or more holds representing the holds that you want to place on the entities.

   For our case study, we select **Investigation of Financial Practices**. Refer to Figure 14-12 on page 375.

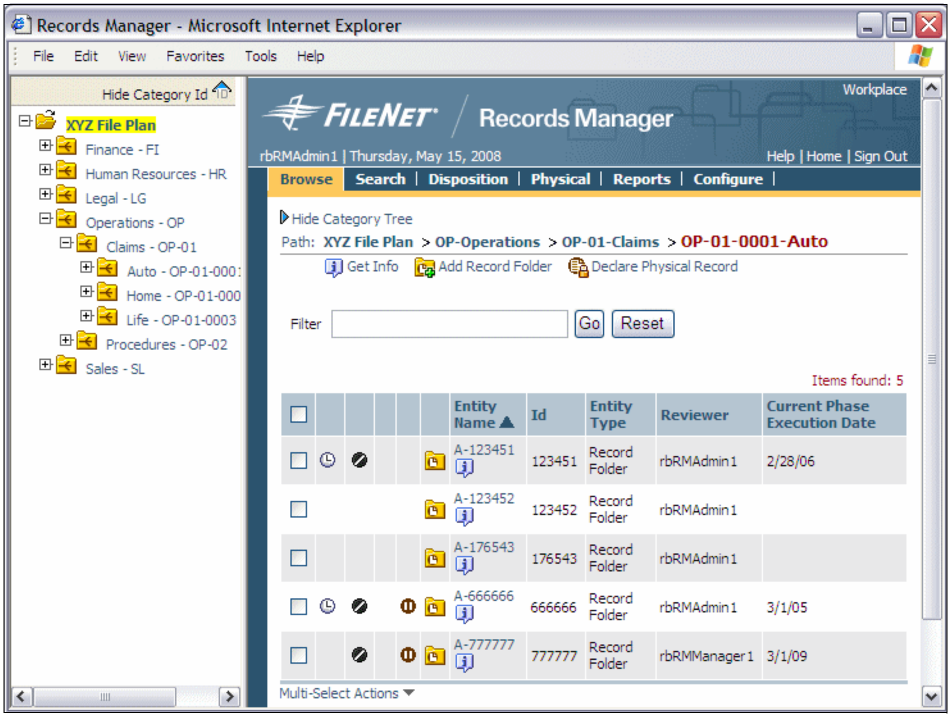

Place Entities On Hold - Microsoft Internet Explorer
File Edit View Favorites Tools Help
FILENET Records Manager
Workplace
rbRMAdmin1 | Thursday, May 15, 2008
Help | Home | Sign Out
Place Entities on Hold
Filter
Go
Reset
Items found: 3
Hold
Exit

□		Entity Name ▲	Hold Reason	Hold Type	Active
☑	●	Investigation of Financial Practices	RedBook case study	Litigation	✓
□	●	Lawsuit with Claims	RedBook case study	Litigation	✓
□	●	Lawsuit with Contractors	RedBook case study	Litigation	✓

Figure 14-12 Selecting hold: Investigation of Financial Practices

5. Click **Hold**, and then, click **OK**.

   For our case study, the result is shown in Figure 14-13 on page 376. Notice that the two records now have the on hold symbols next to their entries.



rbRMAdmin1 | Thursday, May 15, 2008

Path: XYZ File Plan > OP-Operations > OP-01-Claims > OP-01-0001-Auto

Items found: 5

□				Entity Name ▲	Id	Entity Type	Reviewer	Current Phase Execution Date
□	◷	⊘		A-123451	123451	Record Folder	rbRMAdmin1	2/28/06
□				A-123452	123452	Record Folder	rbRMAdmin1	
□				A-176543	176543	Record Folder	rbRMAdmin1	
□	◷	⊘	⦶	A-666666	666666	Record Folder	rbRMAdmin1	3/1/05
□		⊘	⦶	A-777777	777777	Record Folder	rbRMManager1	3/1/09

Figure 14-13 Entities on hold

14.3.2 Removing a hold

After you manually place records on hold, you must manually remove them.

Removing holds manually through individual entities on hold

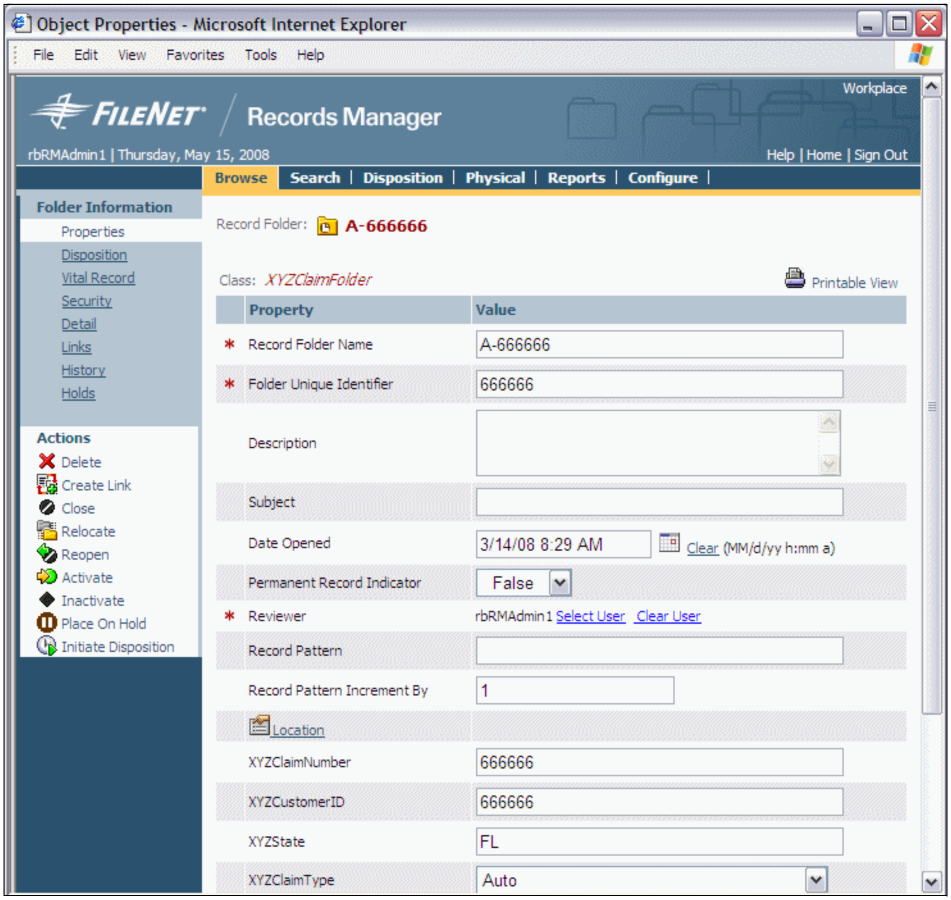
To remove a hold, follow these steps for each entity:

1. Navigate to the entity from which you want to remove the hold.

   For our case study, we navigate to **XYZ File Plan** → **Operations** → **Claims** → **Auto.**

2. Click the **Get Info** icon next to the entity that you want to remove from the hold.

   For our case study, we click the icon next to A-666666. The A-666666 folder information appears. Refer to Figure 14-14 on page 377.



Record Folder: A-666666

Class: XYZClaimFolder

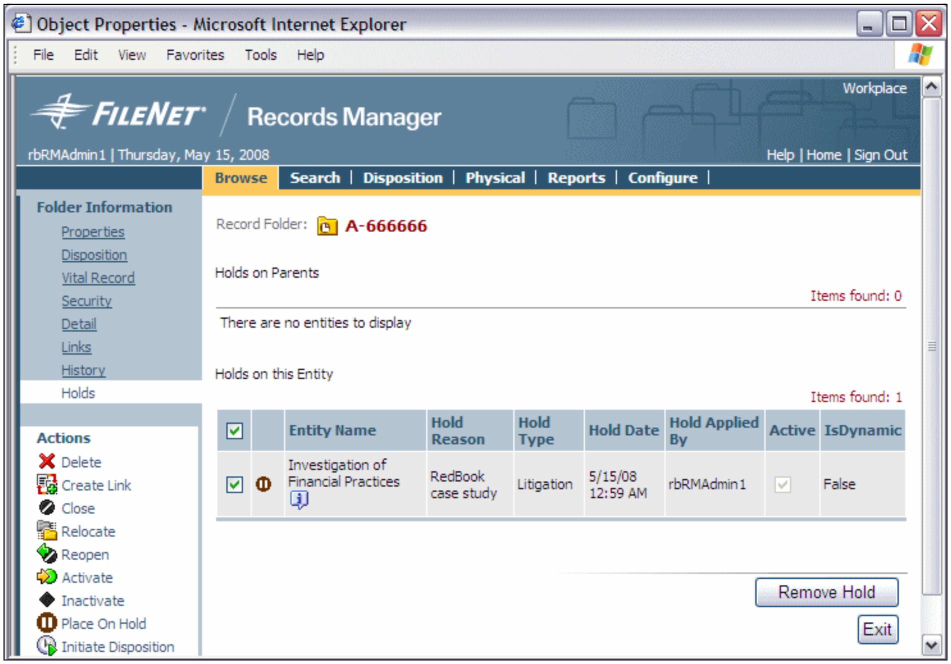
	Property	Value
*	Record Folder Name	A-666666
*	Folder Unique Identifier	666666
	Description	
	Subject	
	Date Opened	3/14/08 8:29 AM Clear (MM/d/yy h:mm a)
	Permanent Record Indicator	False
*	Reviewer	rbRMAdmin1 Select User Clear User
	Record Pattern	
	Record Pattern Increment By	1
	Location	
	XYZClaimNumber	666666
	XYZCustomerID	666666
	XYZState	FL
	XYZClaimType	Auto

Figure 14-14 Entity information page

3. Click **Holds** under Folder Information in the left panel.
4. Select the hold that you want to remove. Refer to Figure 14-15 on page 378.

   It is possible that an entity has multiple holds on it for multiple litigation or various business reasons. In this area, you can remove one or multiple holds on this entity manually.

   For our case study, we have only one hold. Select the check box.



Record Folder: A-666666

Holds on Parents

Items found: 0

There are no entities to display

Holds on this Entity

Items found: 1

☑		Entity Name	Hold Reason	Hold Type	Hold Date	Hold Applied By	Active	IsDynamic
☑		Investigation of Financial Practices	RedBook case study	Litigation	5/15/08 12:59 AM	rbRMAdmin1	☑	False

Figure 14-15 Selecting the hold to remove

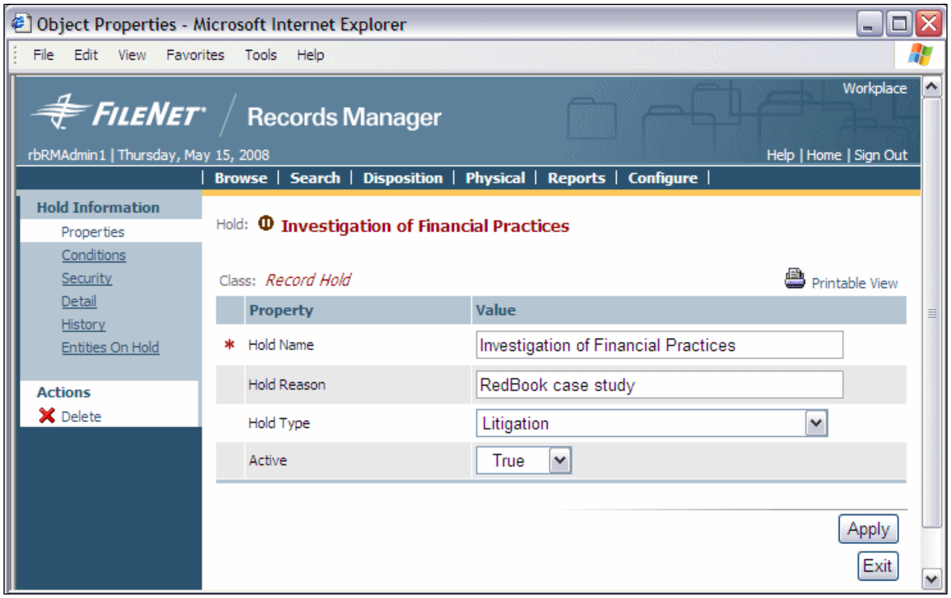
5. Click **Remove Hold**, and click **OK**.

Removing holds by using a hold reason

The manual hold removal instruction that you just reviewed relates to holds on individual entities. For that action, you must know where the entities are located to remove their holds. If you want to remove the holds on multiple entities based on a common hold, follow these steps:

1. Select the **Disposition** tab and click **Holds**.
2. Click the **Get Info** icon of the type of hold that you want to remove.

   For our case study, it is the Investigation of Financial Practices hold. Refer to Figure 14-16 on page 379.


Object Properties - Microsoft Internet Explorer
File Edit View Favorites Tools Help
FILENET Records Manager
Workplace
rbRMAdmin1 | Thursday, May 15, 2008
Help | Home | Sign Out
Browse | Search | Disposition | Physical | Reports | Configure
Hold Information
Properties
Conditions
Security
Detail
History
Entities On Hold
Actions
Delete
Printable View
Apply
Exit

Hold: Investigation of Financial Practices

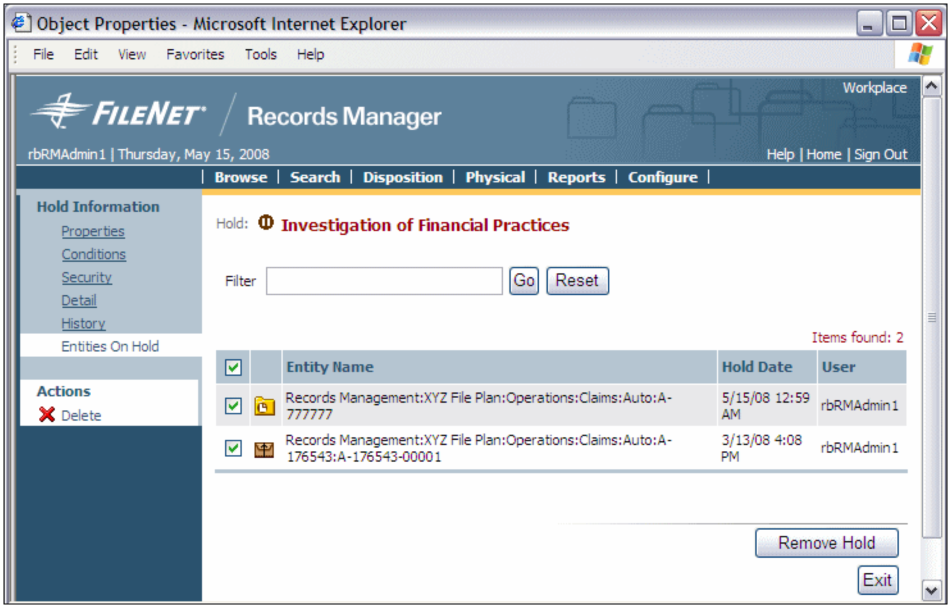
Class: *Record Hold*

	Property	Value
*	Hold Name	Investigation of Financial Practices
	Hold Reason	RedBook case study
	Hold Type	Litigation
	Active	True

Figure 14-16 Hold information page

3. Click **Entities on Hold** in the left panel.

   A list of entities that are placed on this hold appears. Refer to Figure 14-17.


Object Properties - Microsoft Internet Explorer
File Edit View Favorites Tools Help
FILENET Records Manager
Workplace
rbRMAdmin1 | Thursday, May 15, 2008
Help | Home | Sign Out
Browse | Search | Disposition | Physical | Reports | Configure
Hold Information
Properties
Conditions
Security
Detail
History
Entities On Hold
Actions
Delete
Filter
Go
Reset
Remove Hold
Exit

Hold: Investigation of Financial Practices

Items found: 2

☑		Entity Name	Hold Date	User
☑		Records Management:XYZ File Plan:Operations:Claims:Auto:A-777777	5/15/08 12:59 AM	rbRMAdmin1
☑		Records Management:XYZ File Plan:Operations:Claims:Auto:A-176543:A-176543-00001	3/13/08 4:08 PM	rbRMAdmin1

Figure 14-17 Selecting entities that are currently on hold

4. Select the entities from which you want to remove the hold.
5. Click **Remove Hold**, and then, click **OK**.

14.4 Hold Sweep

Hold Sweep is responsible for finding records that meet the conditions that are specified in conditional holds and then placing the holds. In our case study, both the Lawsuit with Contractors and the Lawsuit with Claims holds are dynamic and can be used with Hold Sweep.

Before you can run Hold Sweep, you must configure it for the appropriate values.

14.4.1 Configuring Hold Sweep

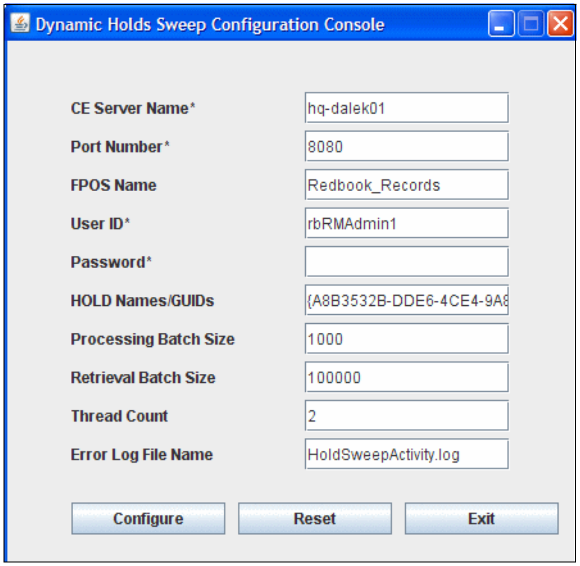
You can configure Hold Sweep to run against a specific File Plan Object Store (FPOS) or all of the FPOSs in a Content Engine. To configure Hold Sweep, perform the following tasks:

1. From a command prompt, navigate to the RecordsManagerSweep folder and execute the following command:

   ```
   RecordsManagerSweep.bat -HoldSweep -configure
   ```

   For our case study, we create a separate directory for the HoldSweep folder. It is under C:\Program Files\FileNet\RM\RedbookRecordsManagerSweep>

   The Dynamic Holds Sweep Configuration Console dialog box appears. Refer to Figure 14-18 on page 381.



CE Server Name*	hq-dalek01
Port Number*	8080
FPOS Name	Redbook_Records
User ID*	rbRMAdmin1
Password*	
HOLD Names/GUIDs	{A8B3532B-DDE6-4CE4-9A8
Processing Batch Size	1000
Retrieval Batch Size	100000
Thread Count	2
Error Log File Name	HoldSweepActivity.log

Figure 14-18 Dynamic Holds Sweep Configuration Console

2. Specify the appropriate values for the following fields:

   – CE Server Name: Name or IP address of the Content Engine server.

     For our case study, it is hq-daleq01.

   – Port Number: The WSI port number that is used by your Content Engine server.

     The default port number for Content Engine running under WebSphere is 9080, WebLogic is 7001, and JBoss is 8080.

     For our case study, we use 8080.

   – FPOS NAME (optional field): Globally Unique Identifier (GUID) or the name of the File Plan Object Store on which you want to run Hold Sweep.

     If you do not provide a value, the Hold Sweep process will run on *all* the File Plan Object Stores that are associated with the specified Content Engine server. If the name of the object store contains extended characters, use the GUID instead of the name.

     For our case study, it is RedBook_Records.

     The GUID is the Globally Unique Identifier (GUID) of the IBM FileNet P8 domain. Every Content Engine object has a GUID that cannot be changed.

   – User ID: The user name that Hold Sweep uses to log on to Content Engine to perform calculations.

  The user must have object store administrative rights on the FPOS and possess Records Administrator privileges.

  For our case study, we use rbRMAdmin1.

 - Password: Password for the user ID.
 - Hold Names/GUIDs: The name or GUID of up to five holds, separated by the '|' character.

   The Hold Sweep process uses only the specified holds. If no holds are provided, the Hold Sweep processes *all the active holds*.

   For our case study, we enter Lawsuit with Contractors|Lawsuit with Claims.

 - Processing Batch Size: The number of entities to be processed as a batch using the Hold Sweep process.

   By default, this value has been set to 1000. For example, if this value is 1000 and there are 20,000 entities to be processed, Hold Sweep will process all entities in 20 batches, with 1000 entities in each batch.

   For our case study, we use the default.

 - Retrieval Batch Size: The number of entities to be retrieved per batch using the Hold Sweep process.

   By default, this value has been set to 100000. For example, if this value is 100000 and there are 1,000,000 entities to be processed, all the entities will be retrieved in 10 batches, with 100000 entities in each batch.

   For our case study, we use the default.

 - Thread Count: The number of threads to be used for hold processing. Typically, this value matches the number of processors on the server where the Hold Sweep is running, but the value can be adjusted based on the tuning of the system.
 - Error Log File Name: The name and path of the error file to be created by the Hold Sweep process, or you can accept the default.

   By default, a file called HoldSweepActivity.log is created in the ../FileNet/RecordsManagerSweep folder.

   For our case study, we use the default.

3. Click **Configure**. You see a message indicating the successful configuration of Hold Sweep.

14.4.2 Running Hold Sweep

Assuming that you have created an active hold, you can automatically place records on hold by running Hold Sweep.

To automatically place records on hold:

1. Run Hold Sweep. From a command prompt, navigate to the RecordsManagerSweep folder and execute the following command:

   ```
   RecordsManagerSweep.bat -HoldSweep
   ```

2. Check the results. Hold Sweep finds records that meet the conditions that were specified in the conditional holds and places them on hold.

Verifying the records that are placed on hold

To verify that the records are held as a result of running Hold Sweep:

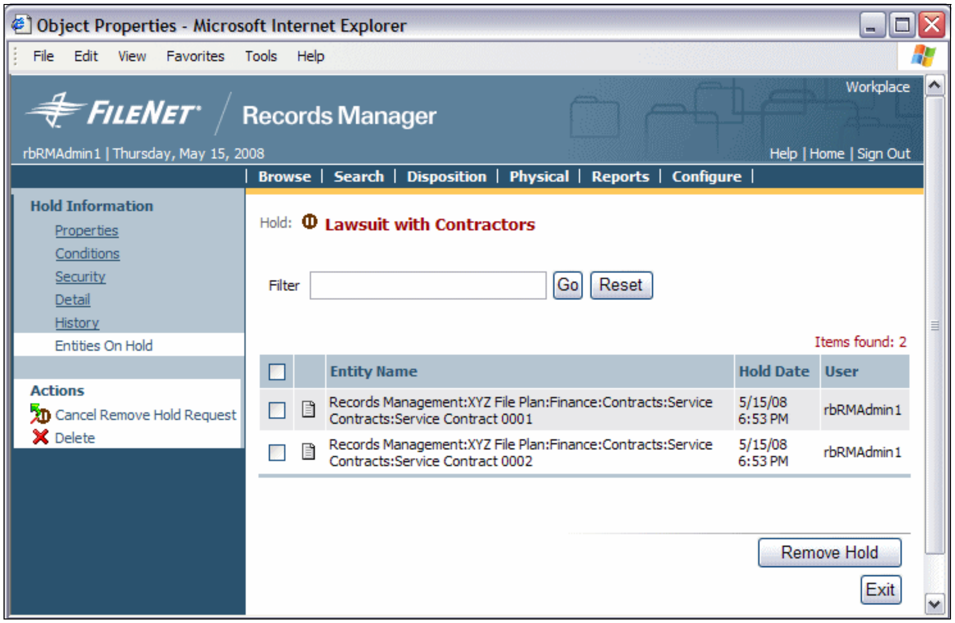
1. Launch IBM FileNet Records Manager.
2. Select the **Disposition** tab. Click **Holds**.
3. Click the **Get Info** icon of the active hold against which you ran Hold Sweep earlier.

   For our case study, we ran against Lawsuit with Contractors Hold.

4. Click **Entities on Hold** in the left panel. Records that are placed on this hold appear.

   For our case study, we have two records on hold. Refer to Figure 14-19 on page 384:

   - Records Management:XYZ File Plan:Finance:Contracts:Service Contracts:Service Contract 0001
   - Records Management:XYZ File Plan:Finance:Contracts:Service Contracts:Service Contract 0002



rbRMAdmin1 | Thursday, May 15, 2008

Hold: Lawsuit with Contractors

Items found: 2

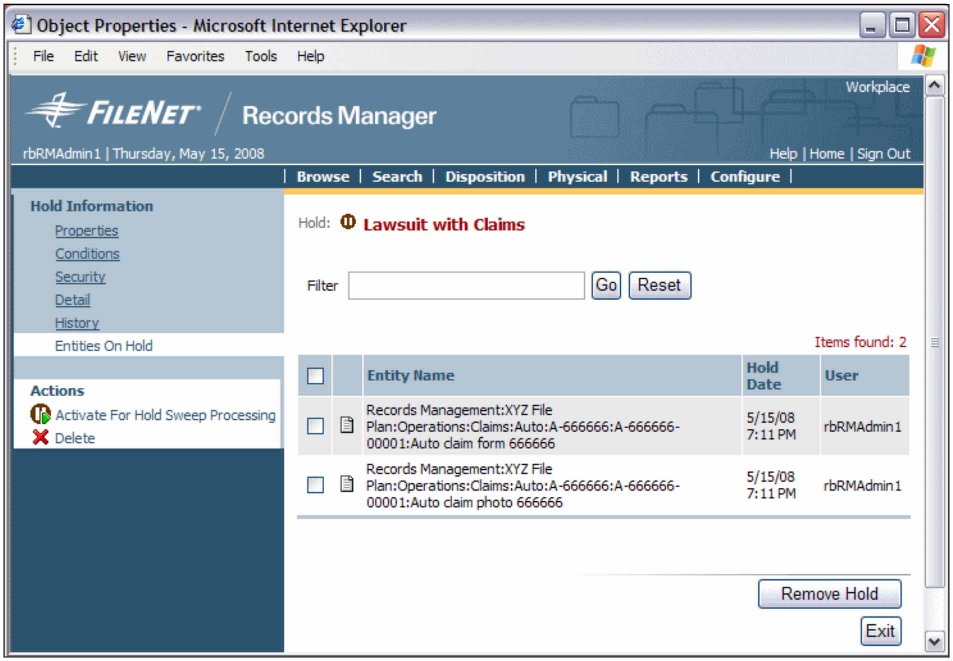
□		Entity Name	Hold Date	User
□		Records Management:XYZ File Plan:Finance:Contracts:Service Contracts:Service Contract 0001	5/15/08 6:53 PM	rbRMAdmin1
□		Records Management:XYZ File Plan:Finance:Contracts:Service Contracts:Service Contract 0002	5/15/08 6:53 PM	rbRMAdmin1

Figure 14-19 Hold Sweep result: Entities on hold for Lawsuit with Contractors Hold

5. Repeat the previous steps for checking the records that are placed under the other hold that you configured for Hold Sweep.

   For our case study, we ran Hold Sweep against both the Lawsuit with Contractors Hold and the Lawsuit with Claims Hold. Hold Sweep identified the following record to be on hold for Lawsuit with Claims Hold (refer to Figure 14-20 on page 385):

   – Records Management:
     XYZ File Plan:Operations:Claims:Auto:A-666666:A-666666-00001:
     Auto claim photo 666666



Object Properties - Microsoft Internet Explorer
File Edit View Favorites Tools Help
FILENET Records Manager
Workplace
rbRMAdmin1 | Thursday, May 15, 2008
Help | Home | Sign Out
| Browse | Search | Disposition | Physical | Reports | Configure |
Hold Information
Properties
Conditions
Security
Detail
History
Entities On Hold
Actions
Activate For Hold Sweep Processing
Delete
Filter
Go
Reset
Remove Hold
Exit

Hold: Lawsuit with Claims

Items found: 2

□		Entity Name	Hold Date	User
□		Records Management:XYZ File Plan:Operations:Claims:Auto:A-666666:A-666666-00001:Auto claim form 666666	5/15/08 7:11 PM	rbRMAdmin1
□		Records Management:XYZ File Plan:Operations:Claims:Auto:A-666666:A-666666-00001:Auto claim photo 666666	5/15/08 7:11 PM	rbRMAdmin1

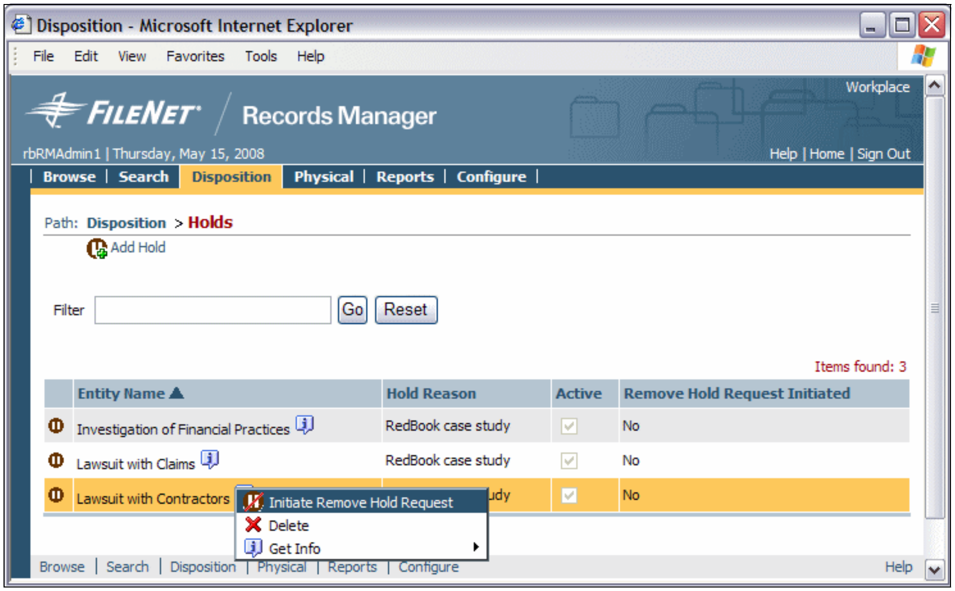
Figure 14-20 Hold Sweep result on second hold: Lawsuit with Claims Hold

14.4.3 Remove dynamic holds using Hold Sweep

When hold is no longer required, you initiate a Remove Hold Request and then run Hold Sweep, using the same name of the hold.

To remove a hold from entities:

1. Launch IBM FileNet Records Manager.
2. Select the **Disposition** tab. Click **Holds**.
3. Right-click the hold, and select **Initiate Remove Hold Request** from the context menu. Refer to Figure 14-21 on page 386.

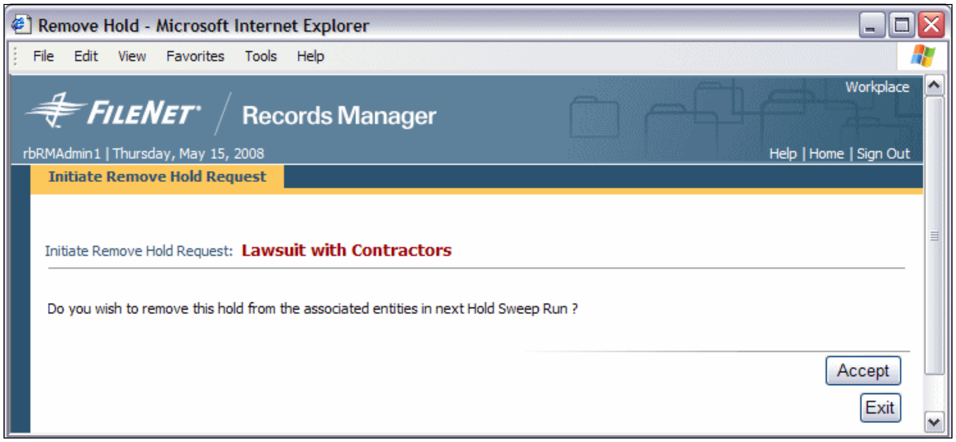

Disposition - Microsoft Internet Explorer
File Edit View Favorites Tools Help
FileNet Records Manager
Workplace
rbRMAdmin1 | Thursday, May 15, 2008
Help | Home | Sign Out
Browse | Search | Disposition | Physical | Reports | Configure
Path: Disposition > Holds
Add Hold
Filter
Go
Reset
Items found: 3
Initiate Remove Hold Request
Delete
Get Info
Browse | Search | Disposition | Physical | Reports | Configure
Help

Entity Name ▲	Hold Reason	Active	Remove Hold Request Initiated
Investigation of Financial Practices	RedBook case study	✓	No
Lawsuit with Claims	RedBook case study	✓	No
Lawsuit with Contractors	udy	✓	No

Figure 14-21 Hold Sweep: Initiate Remove Hold Request

4. Click **Accept** when the system prompts you if you want to remove this hold from associated entities in the next Hold Sweep run. Refer to Figure 14-22.

   If you change your mind, click **Exit** to close the window without initiating the request.

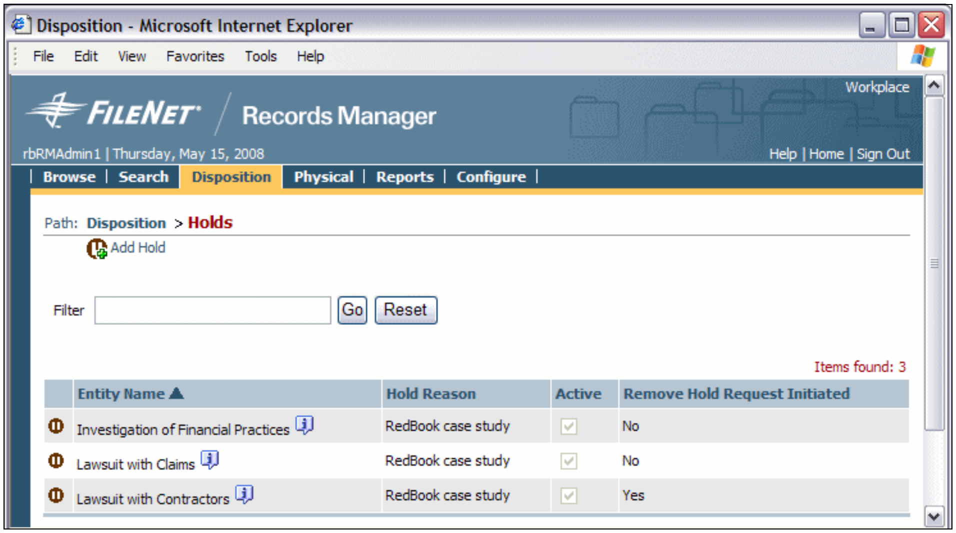

Remove Hold - Microsoft Internet Explorer
File Edit View Favorites Tools Help
FileNet Records Manager
Workplace
rbRMAdmin1 | Thursday, May 15, 2008
Help | Home | Sign Out
Initiate Remove Hold Request
Accept
Exit

Initiate Remove Hold Request: **Lawsuit with Contractors**

Do you wish to remove this hold from the associated entities in next Hold Sweep Run ?

Figure 14-22 Hold Sweep: Initiate Remove Hold Request: Confirm the action

If you click **Accept**, the Remove Hold Request Initiated field will be changed to Yes for the particular hold. For our case study, refer to Figure 14-23.


Disposition - Microsoft Internet Explorer
File Edit View Favorites Tools Help
FILENET
Records Manager
Workplace
rbRMAdmin1 | Thursday, May 15, 2008
Help | Home | Sign Out
Browse | Search | Disposition | Physical | Reports | Configure
Path: Disposition > Holds
Add Hold
Filter
Go
Reset
Items found: 3

Entity Name ▲	Hold Reason	Active	Remove Hold Request Initiated
Investigation of Financial Practices	RedBook case study	☑	No
Lawsuit with Claims	RedBook case study	☑	No
Lawsuit with Contractors	RedBook case study	☑	Yes

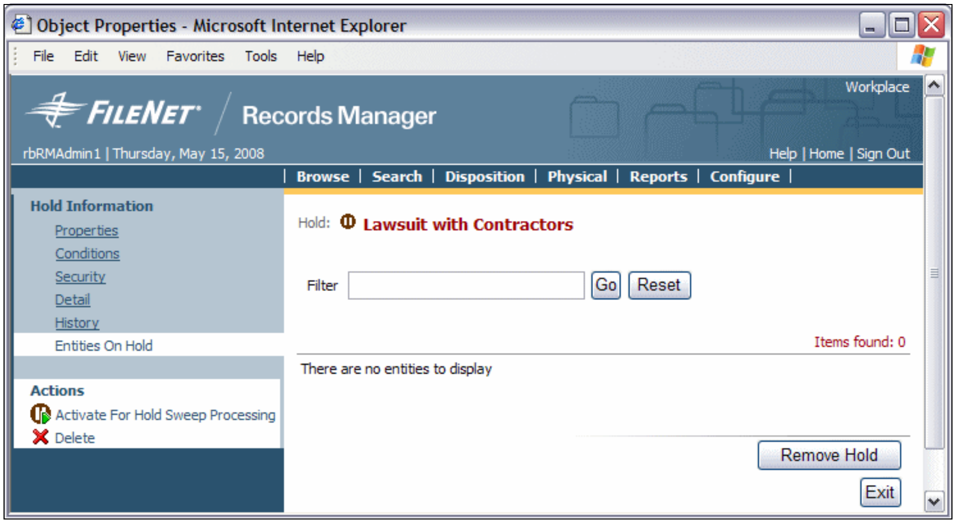
Figure 14-23 Hold Sweep: Remove Hold Request Initiated

5. Run Hold Sweep again using the following command in a command prompt so that the system removes the hold from the corresponding entities:

```
RecordsManagerSweep.bat -HoldSweep
```

6. Verify the results. Hold Sweep finds records that are on hold due to the specific hold (for our case study, Lawsuit with Contractors) and removes the holds from the records.

   To see the result as shown in Figure 14-24 on page 388, go to **Disposition** → **Holds** → **Get Info** (on the hold in question) → **Entities on Hold**. For detailed instructions, refer to “Verifying the records that are placed on hold” on page 383.


Object Properties - Microsoft Internet Explorer
File Edit View Favorites Tools Help
FILENET Records Manager
Workplace
rbRMAdmin1 | Thursday, May 15, 2008
Help | Home | Sign Out
| Browse | Search | Disposition | Physical | Reports | Configure |
Hold Information
Properties
Conditions
Security
Detail
History
Entities On Hold
Actions
Activate For Hold Sweep Processing
Delete
Go
Reset
Remove Hold
Exit

Hold: Lawsuit with Contractors

Filter

Items found: 0

There are no entities to display

Figure 14-24 Hold Sweep: Remove Hold Verification

15

IBM FileNet Records Manager Java APIs case study

In this chapter, we describe how to use the IBM FileNet Records Manager Application Programming Interface (API) for our case study.

We discuss the following topics in this chapter:

- API development environment setup
- Case study

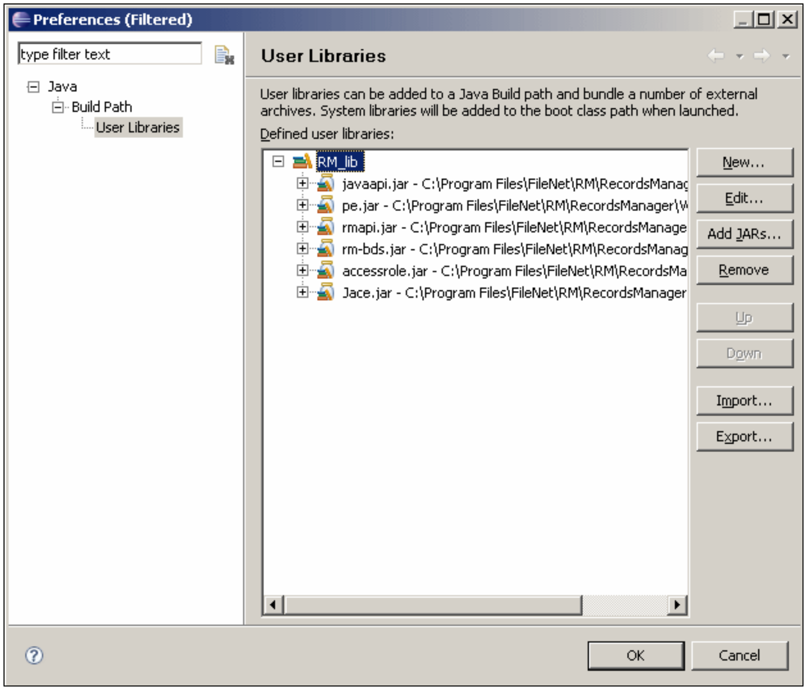
15.1 API development environment setup

In this section, we provide step-by-step instructions to set up the development environment for IBM FileNet Records Manager Java API. For our case study, we use Eclipse Software Development Kit (SDK) 3.3.1.1. We assume that you already have a working IBM FileNet Records Manager environment.

> **Note**: The IBM strategy is to use Rational® Application Development (RAD) to perform Java application development. However, at the time of the writing, we already have a working Eclipse environment and thus document the environment setup using Eclipse.

To set up the development environment, follow these steps:

1. Launch Eclipse (`eclipse.exe`).
2. Open the default Workpace or provide a new file path for your Workpace.
3. Create a new project in eclipse:
   a. Select **File** → **New** → **Java Project**.
   b. Provide the project name in the new project window, and click **Next**.
   c. Leave the default settings for Source. Click the **Library** tab.
   d. Click **Add Library**.
   e. Select **User Library** and click **Next**.
   f. Click **User Libraries** and click **New**.
   g. Click **Next**.
   h. Provide the Library Name and click **OK**.
   i. Click **Add JARs**.
   j. Browse to the lib directory and select the .jar files to include. Select the .jar files as shown in the figure 8-4.



Preferences (Filtered)

type filter text

- Java
  - Build Path
    - User Libraries

User Libraries

User libraries can be added to a Java Build path and bundle a number of external archives. System libraries will be added to the boot class path when launched.

Defined user libraries:

- RM_lib
  - javaapi.jar - C:\Program Files\FileNet\RM\RecordsManag
  - pe.jar - C:\Program Files\FileNet\RM\RecordsManager\W
  - rmapi.jar - C:\Program Files\FileNet\RM\RecordsManage
  - rm-bds.jar - C:\Program Files\FileNet\RM\RecordsManag
  - accessrole.jar - C:\Program Files\FileNet\RM\RecordsMa
  - Jace.jar - C:\Program Files\FileNet\RM\RecordsManager

Figure 15-1 Including .jar files

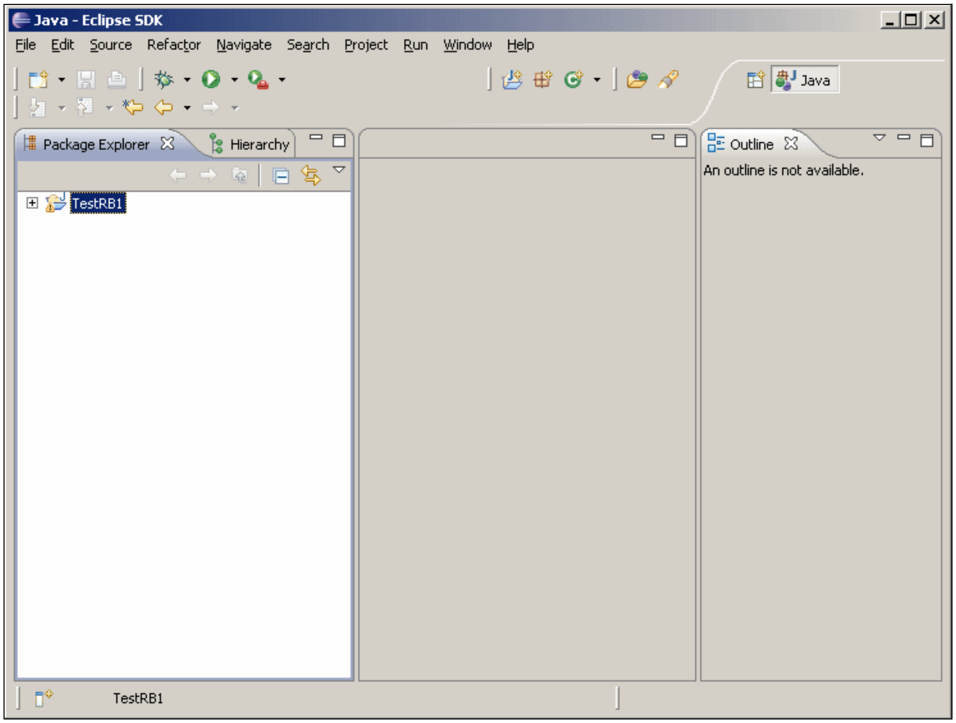
\Program Files\FileNet\RM\RecordsManager\WEB-INF\lib is the default .jar file path. All IBM FileNet Records Manager API .jar files are available at this path.

Include only the six .jar files shown in Figure 15-1, which are sufficient to run a simple IBM FileNet Records Manager API stand-alone program that is provided in the last step. You might be required to include additional .jar files when you develop your custom applications.

k. Click **OK**.

l. Click **Finish** on the Java settings window.

Figure 15-2 on page 392 shows the new Java project that we created for our case study in the project explorer of Eclipse.



An outline is not available.

Figure 15-2 A new Java project created in Eclipse

4. Add the package to your project:
   a. Expand your project name. src shows under the project name.
   b. Right-click **src** and select **Add package** from the context menu.
   c. Enter the package name and click **Finish**. For our case study, we use the default package.
5. Add a new class to your package:
   a. Right-click your package name in the eclipse project explorer and select **Add new class** from the context menu.
   b. Enter the name and click **Finish**. For our scenario, we enter Sample.
   c. Insert the code shown in Example 15-1 on page 393 in the class.
   d. Replace the following information in the code based on your environment's setup:
      - User name (`userName`)
      - Password (`password`)
      - Object store name (`objectStoreName`)

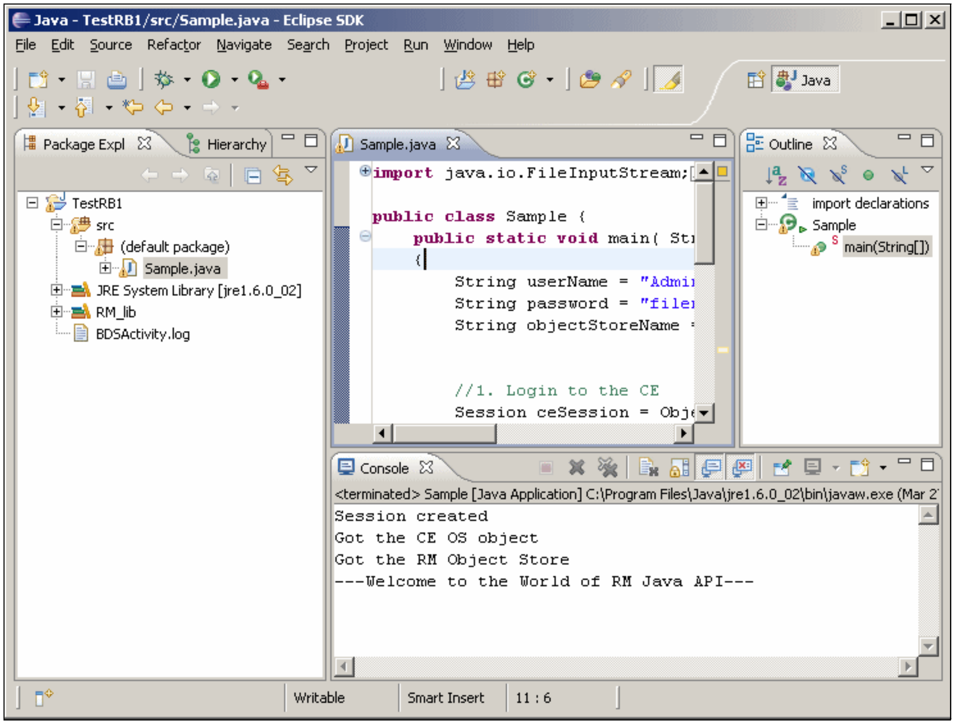
- WcmApiConfig.properties file’s full path (`FileInputStream(“...”)`)

Example 15-1 Sample class code

```
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import com.filenet.rm.api.RMObjectStore;
import com.filenet.rm.api.util.RMUtil;
import com.filenet.wcm.api.ObjectFactory;
import com.filenet.wcm.api.ObjectStore;
import com.filenet.wcm.api.Session;

public class Sample
{
public static void main( String [] args )
    {String userName = "Administrator";
//Replace Administrator with the your CE user name.
    String password = "filenet";
//Replace password with your CE password.
    String objectStoreName = "RBBASEFPOS";
//Replace RBBASEFPOS with your RM object store.

      //1. Login to the CE
        Session ceSession = ObjectFactory.getSession("RM", null, 
userName, password);
        try{
         ceSession.setConfiguration(new FileInputStream("C:\\Program 
Files\\FileNet\\AE\\Workplace\\WEB-INF\\WcmApiConfig.properties"));
//Specify the WcmApiConfig.properties path according to your FileNet 
application installation.
         }catch(FileNotFoundException fe){}
      System.out.println("Session created");

        //2. get the CE ObjectStore ID
       ObjectStore ceObjectStore = null;
        ceObjectStore = ObjectFactory.getObjectStore(objectStoreName, 
ceSession);
        System.out.println("Got the CE OS object");

        //3. get the Records Manager ObjectStore
      RMUtil rmutility = new RMUtil();
      RMObjectStore loRMObjectStore = 
rmutility._getRMObjectStore(ceObjectStore);
       System.out.println("Got the RM Object Store");
      System.out.println("---Welcome to the World of RM Java API---");

    }
}
```

6. From the main menu, select **Run** → **Run**.

   Figure 15-3 shows the output in the console window.



```
import java.io.FileInputStream;[

public class Sample {
    public static void main( Str
    {
        String userName = "Admi
        String password = "filer
        String objectStoreName =


        //1. Login to the CE
        Session ceSession = Obje
```

```
<terminated> Sample [Java Application] C:\Program Files\Java\jre1.6.0_02\bin\javaw.exe (Mar 2
Session created
Got the CE OS object
Got the RM Object Store
---Welcome to the World of RM Java API---
```

Figure 15-3 Sample program output

Now, you can modify the sample program according to your specific need.

15.2 Case study

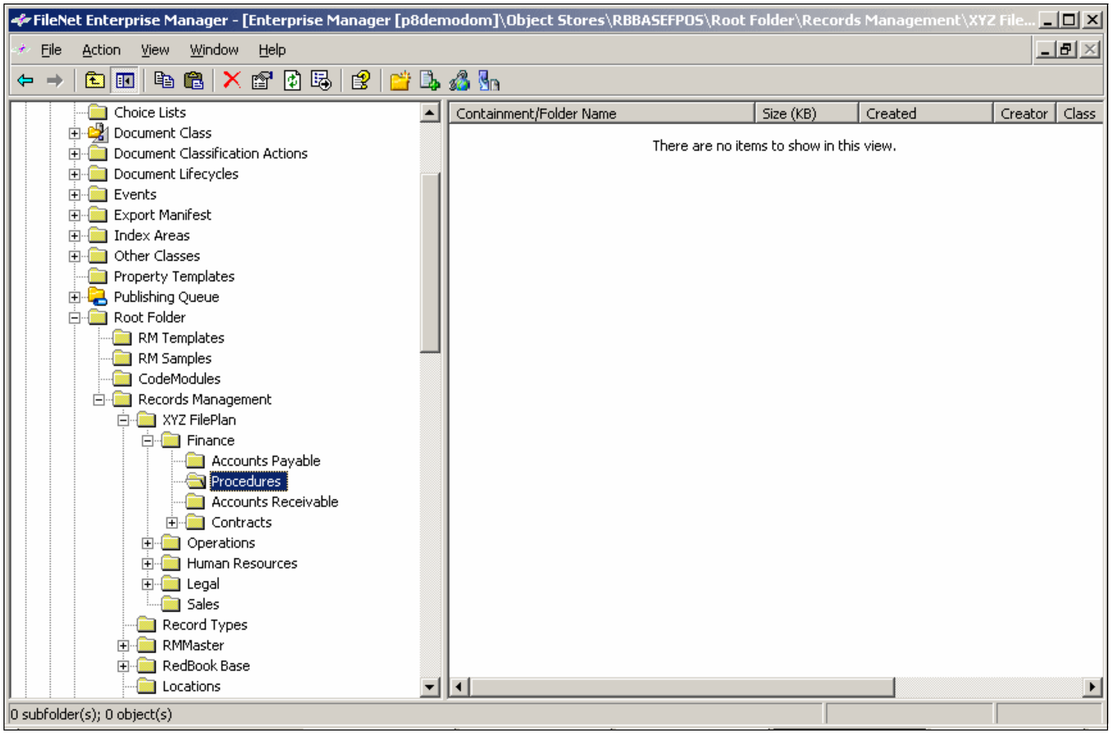
For our case study, we have RBBASEFPOS as the File Plan Object Store (FPOS). In the RBBASEFPOS, we have XYZ FilePlan and the following five departments (categories):

- Finance
- Operations

- Human Resources
- Legal
- Sales

To review the XYZ FilePlan, refer to 3.3.2, “Example file plan” on page 65.

Figure 15-4 shows the file plan setup in IBM FileNet Enterprise Manager. Each of the five departments (categories) is subgrouped into different categories based on the types of records. The procedure category stores the procedure records, and it is available in all five departments.



FileNet Enterprise Manager - [Enterprise Manager [p8demodom]\Object Stores\RBBASEFPOS\Root Folder\Records Management\XYZ File...

Containment/Folder Name	Size (KB)	Created	Creator	Class

There are no items to show in this view.

0 subfolder(s); 0 object(s)

Figure 15-4 File plan structure

15.2.1 Autodeclare records to the categories based on metadata

To show you how to autodeclare records to the specific categories based on specific metadata, we use Procedure documents in our case study as an example. In this example, we want to declare all the Procedure documents as records *automatically* when they come into the IBM FileNet system. Automatic record declaration is done using RMAutoDeclare Content Engine (RMAutoDeclare CE) event. We customize this CE event to implement the automatic procedure of record declaration into the correct department and the correct record class.

All CE events are available inside the directory:

```
\RM Config\Events\
```

To customize the RMAutoDeclare CE event:

1. Modify the RMAutoDeclare.properties file:
   a. Open the property file, which is located at \RM Config\Events\lib directory. Refer to Figure 15-5 on page 397.
   b. Update the *record class* information.

      Specify the targeted record class in this entry. For our case, we want all records to be declared as XYZProcedureRecord records. So, we specify XYZProcedureRecord record class as the target record class:

      ```
      rmevent.declare.RecordClassPropertySymname=XYZProcedureRecord
      ```

      For more flexibility, you can provide a string property as record class entry and enter the desired record class name when a document is added.

   c. Update the *record filed-in folder* information.

      This entry specifies the destination FPOS folder for the record. We specify the XYZDepartment property as the record filed-in folder:

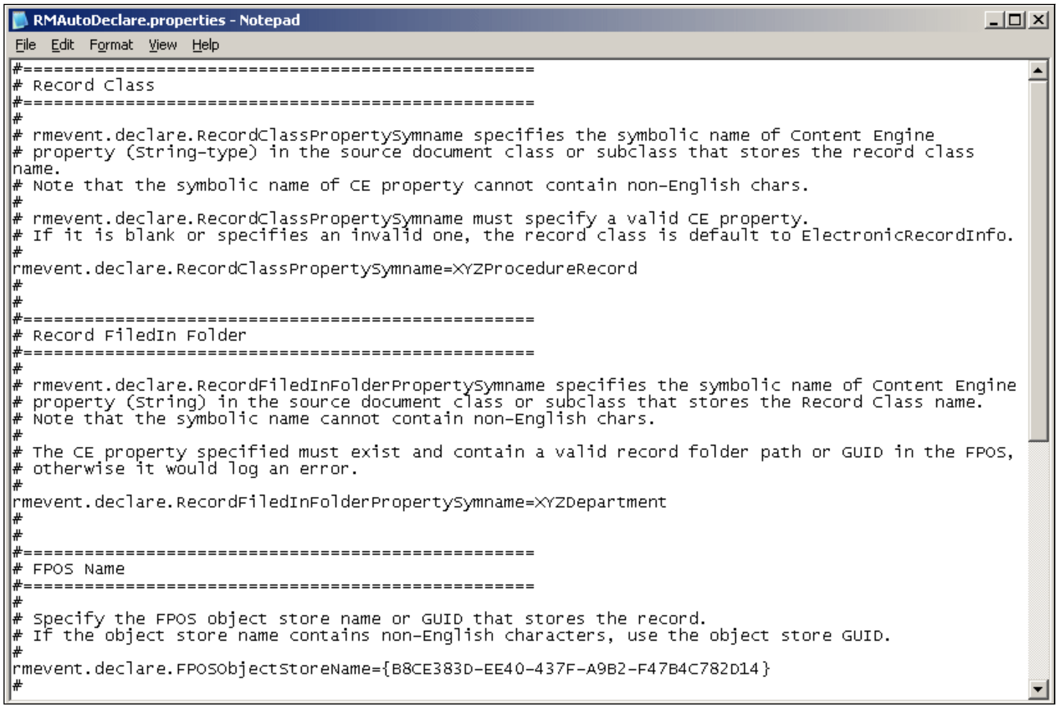
      ```
      rmevent.declare.RecordFiledInFolderPropertySymname=XYZDepartment
      ```

      The XYZDepartment property contains the department name.

   d. Update the *File Plan Object Store* (FPOS) information.

      Specify the FPOS name or FPOS Globally Unique Identifier (GUID) in this entry:

      ```
      rmevent.declare.FPOSObjectStoreName={B8CE383D-EE40-437F-A9B2-F47B
      4C782D14}
      ```


RMAutoDeclare.properties - Notepad
File Edit Format View Help

```
#==================================================
# Record Class
#==================================================
#
# rmevent.declare.RecordClassPropertySymname specifies the symbolic name of Content Engine
# property (String-type) in the source document class or subclass that stores the record class
name.
# Note that the symbolic name of CE property cannot contain non-English chars.
#
# rmevent.declare.RecordClassPropertySymname must specify a valid CE property.
# If it is blank or specifies an invalid one, the record class is default to ElectronicRecordInfo.
#
rmevent.declare.RecordClassPropertySymname=XYZProcedureRecord
#
#
#==================================================
# Record FiledIn Folder
#==================================================
#
# rmevent.declare.RecordFiledInFolderPropertySymname specifies the symbolic name of Content Engine
# property (String) in the source document class or subclass that stores the Record Class name.
# Note that the symbolic name cannot contain non-English chars.
#
# The CE property specified must exist and contain a valid record folder path or GUID in the FPOS,
# otherwise it would log an error.
#
rmevent.declare.RecordFiledInFolderPropertySymname=XYZDepartment
#
#
#==================================================
# FPOS Name
#==================================================
#
# Specify the FPOS object store name or GUID that stores the record.
# If the object store name contains non-English characters, use the object store GUID.
#
rmevent.declare.FPOSObjectStoreName={B8CE383D-EE40-437F-A9B2-F47B4C782D14}
#
```

Figure 15-5 RMAutoDeclare.properties file

2. Update the BDSDeclareContextImpl.java file to use the XYZDepartment property value in the filed-in folder file plan path construction.

   The BDSDeclareContextImpl Java file is available at the following path:

   ```
   \RMConfig\Events\src\com\filenet\rm\ceintegration\eventhandler\declareoperation\impl
   ```

   By default, the following line of code constructs the folder's file plan path:

   `//destFolder = srcDocProps.getStringValue(`*`destFolderProp`*`)`

   Where:

   – *destFolderProp* is mapped to the symbolic property name assigned the filed-in folder `RecordFiledInFolderPropertySymname` in the RMAutoDeclare.properties file.

   For our case study, we construct the file plan path as:

   `XYZ FilePlan/<`*`Department Name`*`>/Procedures`

   We therefore modify our code to:

   ```
   destFolder = "/XYZ FilePlan/" +
   srcDocProps.getStringValue(destFolderProp) + "/Procedures"
   ```

Where:

– Department name is coming from the destFolderProp, which is mapped to the XYZDepartment document property. So, the XYZDepartment property value defines the final file path for the record.

You can use other formula or codes to construct the file plan for the declared record.

3. Compile the BDSDeclareContextImpl.java and add the class file to rm-declare-handler.jar file. This .jar file is located at \RM Config\Events\lib.

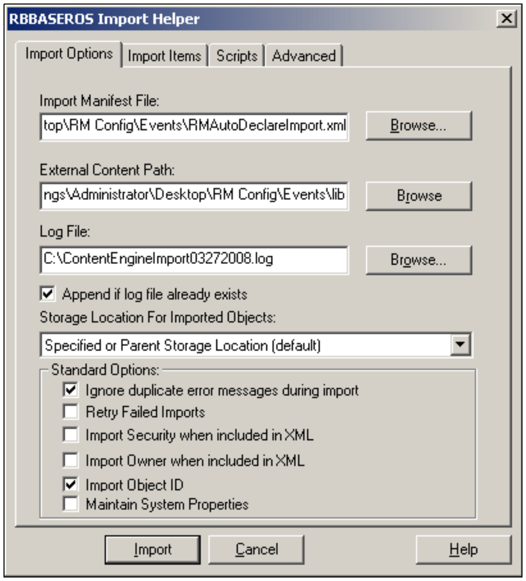
In previous steps, we customize the CE event. Now, we have to implement this customized RMAutoDeclare event in the system.

To implement a customized CE event using IBM FileNet Enterprise Manager, follow these steps:

1. Import RMAutoDeclareImport.xml script on your ROS (RBBASEROS for our case study):

   a. Enter the import file name, RMAutoDeclareImport.xml.

   b. Provide the .jar file's path, \RM Config\Events\lib, for the External Content Path. Figure 15-6 shows the setup for our case study.



Import Manifest File: top\RM Config\Events\RMAutoDeclareImport.xml

External Content Path: ngs\Administrator\Desktop\RM Config\Events\lib

Log File: C:\ContentEngineImport03272008.log

☑ Append if log file already exists

Storage Location For Imported Objects: Specified or Parent Storage Location (default)

Standard Options:

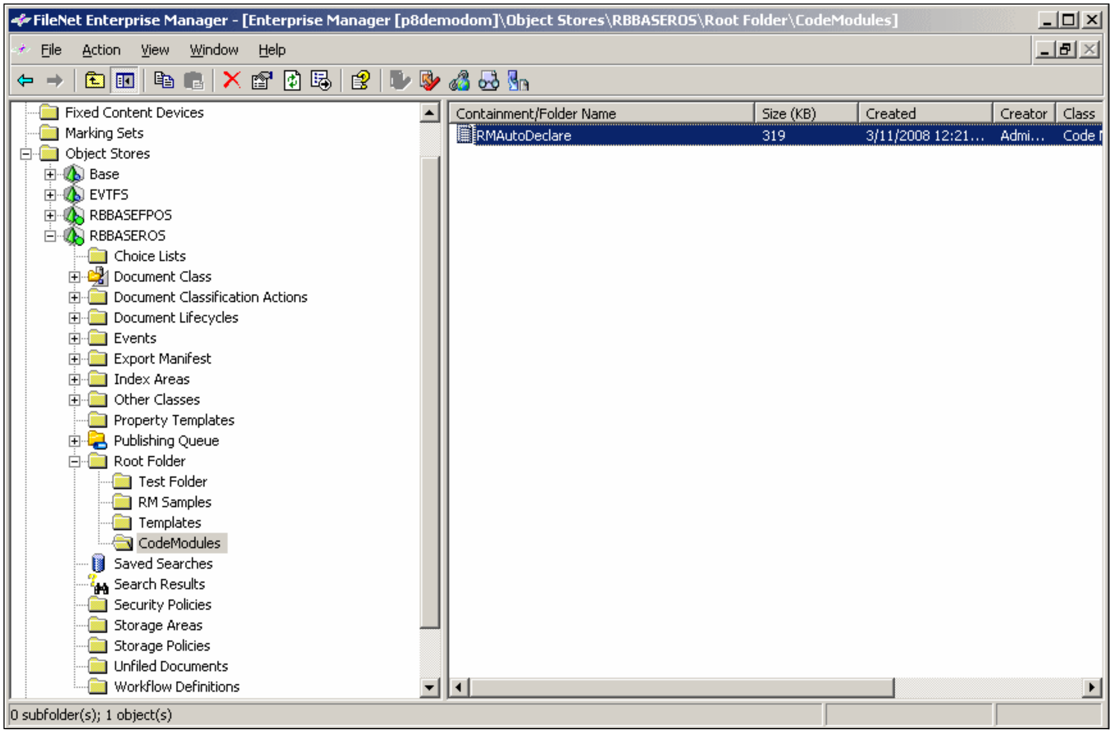
☑ Ignore duplicate error messages during import
☐ Retry Failed Imports
☐ Import Security when included in XML
☐ Import Owner when included in XML
☑ Import Object ID
☐ Maintain System Properties

Figure 15-6 RMAutoDeclareImport.xml script import

   c. Click **Import** and the RMAutoDeclareImport.xml script is imported.

When the AutoDeclare script import finishes, a code module, RMAutoDeclare, is created inside the Code Modules folder. Refer to Figure 15-7. Note this folder can only be viewed from IBM FileNet Enterprise Manager. It is hidden from the Workplace view.

An event action, RMAutoDeclare, is also created after the AutoDeclare script is imported.


FileNet Enterprise Manager - [Enterprise Manager [p8demodom]\Object Stores\RBBASEROS\Root Folder\CodeModules]
File
Action
View
Window
Help

- Fixed Content Devices
- Marking Sets
- Object Stores
  - Base
  - EVTFS
  - RBBASEFPOS
  - RBBASEROS
    - Choice Lists
    - Document Class
    - Document Classification Actions
    - Document Lifecycles
    - Events
    - Export Manifest
    - Index Areas
    - Other Classes
    - Property Templates
    - Publishing Queue
    - Root Folder
      - Test Folder
      - RM Samples
      - Templates
      - CodeModules
    - Saved Searches
    - Search Results
    - Security Policies
    - Storage Areas
    - Storage Policies
    - Unfiled Documents
    - Workflow Definitions

Containment/Folder Name	Size (KB)	Created	Creator	Class
RMAutoDeclare	319	3/11/2008 12:21...	Admi...	Code

0 subfolder(s); 1 object(s)

Figure 15-7 AutoDeclare code module

2. Create a subscription on XYZProcedureDocument class with the *Checkin* event trigger and *RMAutoDeclare* event action.

Now, when a document of class XYZProcedureDocument is added to the system, the Checkin trigger will be triggered and the RMAutoDeclare event action will be executed. When the RMAutoDeclare event action is executed, it will create a record in the FPOS that corresponds to the document that is added into the system.

> **Note:** The RMAutoDeclare event will be triggered whether the document is added to the system automatically or manually by a user.

A

Additional material

This book refers to additional material that can be downloaded from the Internet as described next.

Locating the Web material

The Web material associated with this book is available in softcopy on the Internet from the IBM Redbooks publications Web server. Point your Web browser at:

`ftp://www.redbooks.ibm.com/redbooks/SG247623`

Alternatively, you can go to the IBM Redbooks Web site at:

ibm.com/redbooks

Select the **Additional materials** and open the directory that corresponds with the IBM Redbooks publication form number, SG247623.

Using the Web material

The additional Web material that accompanies this book includes the following files:

File name	*Description*
SG247623.zip	Sample data to be used with case study

System requirements for downloading the Web material

The following system configuration is recommended:

Hard disk space:	40 MB Minimum
Operating System:	Windows
Processor:	Pentium® IV
Memory:	512 MB Minimum

How to use the Web material

Create a subdirectory (folder) on your workstation, and unzip the contents of the Web material zip file into this folder.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks publications

For information about ordering these publications, refer to “How to get IBM Redbooks publications” on page 404. Note that the documents referenced here might be available in softcopy only:

- *Working with IBM Records Manager,* SG24-7389
- *Quick Reference: Records Management 101*, TIPS-0595

  http://www.redbooks.ibm.com/abstracts/tips0595.html?Open

  This technote serves as a quick reference guide for the basic terms and concepts used in the records management field.

Other publications

Additional books include:

- *IBM FileNet P8 Performance Tuning Guide*, GC31-5483

Online resources

These Web sites are also relevant as further information sources:

- IBM FileNet Records Manager product documentation

  http://www.ibm.com/support/docview.wss?rs=3286&context=SSNVVQ&uid=swg27010387

  This Web site contains product release information, product documentation, and technical notices.
- IBM Content Collector information

  http://www.ibm.com/software/data/content-management/content-collector
- IBM Classification Module product information

http://www.ibm.com/software/data/enterprise-search/classification

- IBM eDiscovery product information (eDiscovery Analyzer and eDiscovery Manager):

  http://www.ibm.com/software/data/content-management/products/ediscovery

- IBM FileNet P8 Platform

  http://www.ibm.com/software/data/content-management/filenet-p8-platform

- IBM FileNet Content Manager

  http://www.ibm.com/software/data/content-management/filenet-content-manager

- IBM FileNet Business Process Manager

  http://www.ibm.com/software/data/content-management/filenet-business-process-manager

- IBM FileNet Content Federation Services, refer to:

  http://www.ibm.com/software/data/content-management/filenet-content-manager/federation.html?S_CMP=rnav

- Compliance certificate for IBM FileNet Records Manager 4.5.0

  http://jitc.fhu.disa.mil/recmgt/ibm/filenet/index.html

- ecm_help is the online searchable help that is installed with IBM FileNet Content Manager, and it resides on a J2EE application server. You connect to it by pointing your browser to the server with this URL:

  http://*<host:port>*/ecm_help

How to get IBM Redbooks publications

You can search for, view, or download IBM Redbooks publications, Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy IBM Redbooks publications, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

Understanding IBM FileNet Records Manager

(0.5” spine)
0.475”<->0.875”
250 <-> 459 pages

Understanding IBM FileNet Records Manager

Learn about retention schedules and file plans

Understand how to declare and dispose of records

Review security, audit, hold, and much more

Records management helps users address evolving governance mandates to meet regulatory, legal, and fiduciary requirements. Proactive adherence to information retention policies and procedures is a critical facet of any compliance strategy. IBM FileNet Records Manager helps organizations enforce centralized policy management for file plans, retention schedules, legal preservation holds, and auditing. IBM FileNet Records Manager enables organizations to securely capture, declare, classify, store, and dispose of electronic and physical records.

In this IBM Redbooks publication, we introduce the records management concept and provide an overview of IBM FileNet Records Manager. We address records management topics including retention schedule, file plan, records ingestion and declaration, records disposition, records hold, and IBM FileNet Records Manager APIs. In addition, using a case study, we describe step-by-step instructions to implement a sample records management solution using IBM FileNet Records Manager. We provide concrete examples of how to perform tasks, such as file plan creation, records ingestion and declaration, records disposition, and records hold.

This book is for anyone who wants a basic understanding of the records management concept, the features and capabilities that IBM FileNet Records Manager offers, and its basic usage.

SG24-7623-00 ISBN 0738432571