

Using IBM Enterprise Records

Whei-Jen Chen

Serena S Chan

Jean-Marc Costecalde

Yolanda H Yates

Harry Yessayan



Information Management



International Technical Support Organization

Using IBM Enterprise Records

May 2015

Note: Before using this information and the product that supports, read the information in “Notices” on page xi.

Second Edition (May 2015)

This edition applies to IBM Enterprise Records Version 5.2.

© Copyright International Business Machines Corporation 2009, 2015. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	.xi
Trademarks	xii
IBM Redbooks promotions	xiii
Preface	xv
Authors	xv
Acknowledgements	xvi
Now you can become a published author, too	xvii
Comments welcome	xvii
Stay connected to IBM Redbooks	xviii
Summary of changes	xix
May 2015, Second Edition	xix
Part 1. Concept	1
Chapter 1. Records management	1
1.1 What constitutes a record	2
1.2 What records management involves	4
1.3 The business challenge: Information lifecycle governance	5
1.4 The importance of records management	7
1.5 Legal, regulations, compliance, and investigations	8
1.5.1 Addressing regulatory requirements	10
1.5.2 Investigations	15
1.6 Planning an information lifecycle governance program	16
1.6.1 Obtaining corporate sponsorship and stakeholder buy-in	18
1.7 Records management maturity model	19
1.7.1 Using an objective records management maturity model	19
1.8 Organizational readiness	22
1.9 Records management system technical standards and guidelines	25
1.10 Role of IBM Enterprise Records within the IBM Information Lifecycle Governance portfolio	28
1.10.1 Policy management relationship to IBM Enterprise Records	29
1.10.2 Syndicating global retention and schedule management policies to Enterprise Records	31
Chapter 2. IBM Enterprise Records system and architecture	35
2.1 Overview of IBM Enterprise Records	36

2.1.1	Key business benefits of IBM Enterprise Records	36
2.1.2	Software highlights and capabilities	37
2.1.3	Working with IBM Enterprise Records	38
2.2	System architecture	41
2.2.1	Enterprise Records for IBM Content Foundation	41
2.2.2	Relationship between content and records	46
2.3	Data model, workflow, and security	48
2.3.1	IBM Enterprise Records data model	48
2.3.2	IBM Enterprise Records workflows	51
2.3.3	IBM Enterprise Records security and roles	51
2.4	Logging	53
2.5	User and administrative applications	53
2.5.1	IBM Content Navigator	54
2.5.2	IBM Enterprise Records interface	56
2.5.3	Disposition process	60
2.5.4	Hold process	60
2.5.5	IBM Administration Console for Content Engine	61
2.5.6	File Plan Import Export Tool	61
2.6	APIs and the Component Integrator	61
2.6.1	IBM Enterprise Records and Bulk Declaration Services	61
2.6.2	IBM Enterprise Records Component Integrator	62
2.7	Reporting	63
2.8	References	63
Chapter 3.	Retention and file plans	65
3.1	Retention schedule	66
3.2	Retention schedule planning and creation	67
3.2.1	Develop a records management policy	68
3.2.2	Specify records management procedures	70
3.2.3	Record and update regulatory requirements	71
3.2.4	Conduct a records inventory	71
3.2.5	Define records series	72
3.2.6	Create a regulatory matrix	73
3.2.7	Creating the retention schedule	75
3.3	File plan	79
3.4	File plan planning and creation	82
3.5	File plan in IBM Enterprise Records	83
3.5.1	File plan elements	83
3.5.2	Attributes of containers and records	88
3.6	Case study: File plans in IBM Enterprise Records	90
Chapter 4.	Security	93
4.1	Security model overview	94

4.2	Records management roles and security	95
4.2.1	Four standard roles	96
4.2.2	Roles and access levels	99
4.2.3	Mapping roles to security groups	99
4.3	Determining the security model	102
4.3.1	Security proxy types	103
4.3.2	Containers as security parents	104
4.3.3	Controlling security by full proxy	105
4.3.4	Relating file plan structure to access control	105
4.3.5	Access control that differs from the file plan structure	107
4.4	Individual record security	108
4.4.1	Marking sets	108
4.4.2	Direct security	112
4.4.3	Comparing approaches	112
4.5	Security and record holds	114
4.6	Limiting functional access	114
4.6.1	Limiting access to a desktop	114
4.6.2	Limiting access to features	116
4.6.3	Limiting access to specific functions within a view	118
4.7	Separating records into multiple repositories	118
	Chapter 5. Records capture, creation, and retrieval	119
5.1	Why automation is the goal	120
5.1.1	Successfully automating record creation and capture	121
5.1.2	The complexities of manual record creation and capture	125
5.1.3	Overview of content ingestion and declaration	125
5.2	Record capture	128
5.2.1	Document and record classes	129
5.2.2	Manual declaration	133
5.3	Manual record creation and capture	133
5.3.1	Document entry templates	134
5.3.2	Record entry templates	136
5.3.3	Record classification considerations	137
5.3.4	Primary mechanism for manual ingestion and declaration	142
5.3.5	Working with document versions	147
5.4	Performance considerations	150
	Chapter 6. Records disposition and basic schedules	151
6.1	Introduction to records disposition	152
6.1.1	Importance of records disposition	152
6.2	Implementing records disposition policies	154
6.2.1	Basic disposition	154
6.2.2	Advanced disposition	155

6.2.3	Scheduling and monitoring disposition sweeps	155
6.2.4	Completing the disposition process	156
6.3	Basic disposition schedules	157
6.3.1	Characteristics of basic disposition schedules	157
6.3.2	Creating a basic disposition schedule	160
6.3.3	Converting a record category to a basic schedule	161
6.3.4	Basic disposition sweep and processing	162
6.3.5	Example use cases for basic disposition sweep	167
6.3.6	Controlling how records are grouped for disposition	171
Chapter 7.	Advanced disposition	175
7.1	Advanced disposition schedules	176
7.1.1	Disposition schedule	176
7.1.2	Disposal triggers	179
7.1.3	Cutoff	183
7.1.4	Disposition phases and actions	185
7.1.5	Disposition workflows	190
7.1.6	Alternate retention	191
7.1.7	Assigning disposition schedules to the file plan	191
7.1.8	Record types	198
7.2	Advanced disposition sweep	199
7.2.1	Advanced disposition sweep for disposition processing	199
7.2.2	Setting an advanced disposition sweep to run from the desktop	201
7.2.3	Running an advanced disposition sweep from the desktop	201
7.3	Initiating and completing disposition	202
7.3.1	Initiating disposition manually	203
7.3.2	Initiating a disposition by scheduling a sweep	204
7.3.3	Strategies for initiating disposition	204
7.3.4	Disposition processing in batches	205
7.3.5	Completing the disposition process	206
7.4	Automatic destruction using Auto Destroy	207
7.4.1	When to use Auto Destroy	207
7.4.2	Running Auto Destroy from the desktop	207
7.5	Running a sweep from the command line	208
7.5.1	Configuring an advanced disposition sweep	208
7.5.2	Deployment and scheduling considerations	210
7.6	Performance considerations	211
7.7	Converting advanced schedules to basic schedules	212
Chapter 8.	Holds and preservation	213
8.1	Definition of hold	214
8.2	Hold processing in IBM Enterprise Records	214
8.2.1	Audit and legal holds	216

8.2.2	Manual holds	217
8.2.3	Dynamic holds	217
8.2.4	Multiple holds	218
8.2.5	Applying holds	219
8.2.6	Removing holds	221
8.2.7	Running Hold Sweep	225
8.2.8	Inheritance of holds	231
8.2.9	Disposal trigger aggregation level effect on holds in advanced dispositions	232
8.3	Performance considerations	232
Chapter 9. Audit requirements		235
9.1	Introduction to audits	236
9.1.1	Compliance audits	236
9.1.2	Evidential weight	236
9.2	Audits of an IBM Enterprise Records system	237
9.3	Accessing the audit log	238
9.3.1	Accessing audit information from IBM Enterprise Records	238
9.3.2	Accessing audit information with the Content Platform Engine	239
9.4	Reporting by using the audit data	241
9.5	Pruning the audit log	241
Chapter 10. Reporting		243
10.1	Reporting capabilities and considerations	244
10.2	Running IBM Cognos reports	244
10.2.1	Configuration	245
10.2.2	Predefined reports	245
10.2.3	Running reports from IBM Cognos	246
10.2.4	Adding a new Cognos report	248
10.3	Running reports from Crystal Reports	248
10.3.1	Configuration	248
10.3.2	Predefined reports	249
Chapter 11. Physical records		253
11.1	Overview of physical records management	254
11.2	Enterprise Records physical records capabilities	255
11.2.1	Containers: Boxes	256
11.2.2	Containers: Physical and hybrid folders and folder volumes	261
11.2.3	Bar codes	263
11.2.4	Searching	266
11.2.5	Reporting	272
11.2.6	Auditing	272
11.3	Tracking physical records	272
11.3.1	Workflow subscriptions for physical records management	272

11.3.2	Accessing physical records	273
11.3.3	Locations, reservations, and charge outs	274
Chapter 12. IBM Enterprise Records Java APIs		279
12.1	Introduction to IBM Enterprise Records APIs	280
12.2	Java API for Records Manager	281
12.3	Records Manager API	281
12.4	Bulk Declaration Service	282
12.5	Performance considerations	283
Chapter 13. IBM Enterprise Records for IBM Content Manager		285
13.1	Presentation	286
13.2	Architecture	286
13.3	Difference with Content Federation Services	287
13.4	Java for Records Manager	289
Part 2. Implementation case studies		293
Chapter 14. File plan case study		295
14.1	Types of object stores	296
14.1.1	File plan object store	296
14.1.2	Record-enabled object store	297
14.2	File plan case study introduction	297
14.3	Creating a file plan in IBM Enterprise Records	298
14.3.1	Create a new file plan	299
14.3.2	Browse the file plan	301
14.3.3	Populate the file plan	303
Chapter 15. Basic disposition case study		309
15.1	Create a new record category with a basic disposition schedule	310
15.2	Schedule a basic disposition sweep for report only	313
15.2.1	Schedule the sweep	313
15.2.2	View the sweep results	315
15.3	Schedule a basic disposition sweep for immediate destruction	316
15.3.1	Schedule the sweep	316
15.3.2	View the sweep results	318
15.3.3	Verify the destroy results	319
15.4	Schedule a basic disposition sweep for approval before destruction	321
15.4.1	Schedule the sweep	322
15.4.2	View the sweep results	323
15.4.3	Approve the records for destruction	324
15.4.4	Verify the destroy results	325

Chapter 16. Advanced disposition case study	327
16.1 Configure advanced disposition for approval before destruction	328
16.1.1 Add a Destroy action	328
16.1.2 Add an internal event trigger	330
16.1.3 Add an advanced disposition schedule	331
16.1.4 Assign a disposition schedule to a record category	333
16.2 Schedule and complete advanced disposition	336
16.2.1 Schedule the advanced disposition sweep	336
16.2.2 Monitor and verify the sweep results	339
16.2.3 Initiate disposition by running a sweep	341
16.2.4 Complete the Destroy workflow process	343
16.3 Configure advanced disposition for automatic destruction	347
16.3.1 Add the Auto Destroy action	348
16.3.2 Add the internal event trigger	348
16.3.3 Assign the disposition schedule to the record category	350
16.3.4 Assign the schedule to the correct record category	351
16.4 Schedule and complete advanced disposition for Auto Destroy	352
16.4.1 Schedule the advanced disposition sweep	352
16.4.2 Schedule Auto Destroy	354
16.5 Convert a record category to a basic schedule	356
16.5.1 Identify an eligible record category	356
16.5.2 Schedule the container conversion	357
Chapter 17. Records hold case study	363
17.1 Case study hold scenarios	364
17.2 Creating a hold	364
17.3 Manually placing and removing holds	368
17.3.1 Manually placing an entity on hold	368
17.3.2 Removing a hold	370
17.4 Dynamic holds and Hold Sweep	373
17.4.1 Launching Hold Sweep as a batch process	374
17.4.2 Launching Hold Sweep with Content Navigator Task Manager	376
17.4.3 Hold status changes	378
17.4.4 Verifying the records that are placed on hold	379
17.4.5 Removing dynamic holds using Hold Sweep	380
Chapter 18. IBM Java API for Records Manager case study	385
18.1 Description of the use cases	386
18.1.1 Content Engine class and properties setup	387
18.1.2 Use Case 1 walk-through	389
18.1.3 Use Case 2 walk-through	390
18.2 Record populating batch sample code	392
18.3 Event handler for record maintenance sample code	399

18.3.1 Update the existing AutoSyncProperties.java file	400
Related publications	411
IBM Redbooks	411
Online resources	411
Help from IBM	412

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 grant 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 websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites 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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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

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

COPYRIGHT LICENSE:

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

Trademarks

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:

Cognos®

DB2®


developerWorks®

FileNet®

IBM®

Optim™

Redbooks®

Redbooks (logo) ®

StoredIQ®

WebSphere®

The following terms are trademarks of other companies:

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

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

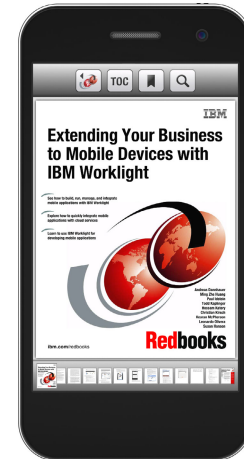
UNIX is a registered trademark of The Open Group in the United States and other countries.

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

Find and read thousands of IBM **Red**books publications

- ▶ Search, bookmark, save and organize favorites
- ▶ Get up-to-the-minute Redbooks news and announcements
- ▶ Link to the latest Redbooks blogs and videos

Get the latest version of the **Red**books Mobile App



Promote your business in an IBM **Red**books publication

Place a Sponsorship Promotion in an IBM® Redbooks® publication, featuring your business or solution with a link to your web site.

Qualified IBM Business Partners may place a full page promotion in the most popular Redbooks publications. Imagine the power of being seen by users who download millions of Redbooks publications each year!



ibm.com/Redbooks

About Redbooks → Business Partner Programs

THIS PAGE INTENTIONALLY LEFT BLANK

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® Enterprise Records helps organizations enforce centralized policy management for file plans, retention schedules, legal preservation holds, and auditing. IBM Enterprise Records 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 Enterprise Records. We address records management topics, including the retention schedule, file plan, records ingestion and declaration, records disposition, records hold, and Enterprise Records application programming interfaces (APIs). We also use a case study to describe step-by-step instructions to implement a sample records management solution using Enterprise Records. 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 helps you to understand the records management concept, the IBM Enterprise Records features and capabilities, and its use.

Authors

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

Whei-Jen Chen is a Project Leader at the International Technical Support Organization, San Jose Center. She has extensive experience in application development, database design and modeling, and IBM DB2® system administration. Whei-Jen is an IBM Certified Solutions Expert in database administration and application development and an IBM Certified IT Specialist.

Serena S Chan is an Associate Partner with IBM Global Business Services' Financial Services Sector in Toronto. She has more than 20 years of consulting experience and has won the prestigious IBM Ovation! Award and the IBM Service Excellence Award on five occasions. Serena is a Project Management Professional (PMP) and holds numerous certifications, including Disciplined Agilist-Yellow Belt, IBM Certified e-Business Solution Designer, and IBM Certified

Solution Developer. She co-authored three other IBM Redbooks publications: *IBM WebSphere Portal for Multiplatforms V5 Handbook*, SG24-6098; *IBM WebSphere Portal V5 A Guide for Portlet Application Development*, SG24-6076; and *Working with IBM Records Manager*, SG24-7389.

Jean-Marc Costecalde is an Enterprise Content Management Quality Assurance specialist in the IBM Software Group, in the United States. He works on different content management products, including IBM Enterprise Records, and has 15 years of experience in the content management field, working in the software development group.

Yolanda H Yates is a Senior Technical Consultant with IBM Software Group in Australia. Yolanda has 23 years of IT experience with an emphasis on process and records consulting. She has a double degree from Latrobe university in Melbourne Australia. Her expertise is IBM Enterprise Content Management products. Before joining IBM in 2006, Yolanda provided training and solution design for IBM FileNet® P8 Content Management, Process Management, Forms Management, and Records Management across the Asia-Pacific region.

Harry Yessayan is a Solution Architect and Managing Consultant for IBM Software Group's Enterprise Content Management Software Services in the United States. He has more than 15 years of experience in enterprise content management, most recently focusing on solutions for records and retention management and information lifecycle governance. He holds a Master of Science degree in Information and Computer Science from the University of California, Irvine. He co-authored the IBM Redbooks publication titled *Creating Value-Based Archiving Solutions with IBM Content Collector*, SG24-8078, and is one of the original authors of the First Edition of this book.

Acknowledgements

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

Ned Bader
William Belknap
Ruen Dineros
Richard Hogg
Stewart Imagawa
Chi Nguyen
Ron Rathgeber
Steven Wilson

IBM Software Group, USA

Dong Rui Li
IBM Software Group, China

Thanks to the authors of the First Edition of this book, *Understanding IBM FileNet Records Manager*, SG24-7623, published in May 2009:

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

Now you can become a published author, too

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time. Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

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

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 Redbooks form:

ibm.com/redbooks

- ▶ Send your comments by email:

redbooks@us.ibm.com

- ▶ Mail your comments:

IBM Corporation, International Technical Support Organization
Dept. HYTD Mail Station P099

2455 South Road
Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- ▶ Find us on Facebook:
<http://www.facebook.com/IBMRedbooks>
- ▶ Follow us on Twitter:
<http://twitter.com/ibmredbooks>
- ▶ Look for us on LinkedIn:
<http://www.linkedin.com/groups?home=&gid=2130806>
- ▶ Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:
<https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm>
- ▶ Stay current on recent Redbooks publications with RSS Feeds:
<http://www.redbooks.ibm.com/rss.html>

Summary of changes

This section describes the technical changes made in this edition of the book and in previous editions. This edition might also include minor corrections and editorial changes that are not identified.

Summary of Changes
for SG24-7623-01
for *Using IBM Enterprise Records*

May 2015, Second Edition

This revision reflects the addition, deletion, or modification of new and changed information.

New information

IBM Enterprise Records Version 5.2 features and functions

Changed information

Use cases and features and functions of IBM Enterprise Records Version 5.2



Part 1

Concept

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



Records management

This chapter provides an overview of records management. We describe the concept of a *record*, the importance of records management, and how it relates to information lifecycle governance.

We cover the following topics:

- ▶ What constitutes a record
- ▶ What records management involves
- ▶ The business challenge: Information lifecycle governance
- ▶ The importance of records management
- ▶ Legal, regulations, compliance, and investigations
- ▶ Planning an information lifecycle governance program
- ▶ Records Management Maturity Model
- ▶ Organizational readiness
- ▶ Records Management System Technical Standards and Guidelines
- ▶ Role of IBM Enterprise Records within IBM Information Lifecycle Governance

1.1 What constitutes a record

A *record* is any type of content that states results achieved, pertains to, and provides evidence of activities performed. There are four essential characteristics of a record:

- ▶ *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 is generally retained for analysis, legal and corporate policy, or historical purposes and as a representation of what occurred. It can be in any type of format, including soft or hard copy.

As depicted in Figure 1-1 on page 3, a record can take the form of paper records, microfiche, electronic documents, email, fax, instant messaging, collaboration content, voice recording, wireless communication content, audio, video, shared drive content, Web content, or documents on cloud storage. Email messages are a common sources of records as a history of discussions and decisions made, which is a primary reason for eDiscovery on email today.



Figure 1-1 Types of records

Records can be any business or personal transaction. Records are often made up of a group of related content, not always a single individual email message, file, or document.

Records can include trade instructions, trade confirmations, articles of incorporation, bylaws, or standard operating procedures. Records can be stored on any medium, such as diskettes, tape, optical disks, and shared drives. Records can be generated internally within a company or can be received from other sources.

Records are similar to other assets of a company. They are valuable and subject to industry regulations. Many countries around the world have legislation related to recordkeeping. Most are applicable to physical and electronic records, some specify the active and inactive retention period, and some have special compliance requirements for *storage media*. Some industries are more heavily regulated than others and some of those regulations are more complex. Conflicting regulations that produce contradictory records retention periods across different jurisdictions or locations where they operate are problems for organizations.

Historically, the simplest path to take was to pick the longest imposed retention period across all relevant content in the business, worldwide. However, this has the potential to lead to over-retention, unnecessary costs for storage and related costs, and far greater eDiscovery review costs. A modern information governance program that implements a modern records retention program is able to identify and deal with these differences and ensure that only the relevant records content is preserved for the minimal amount of time for each jurisdiction.

To illustrate, we review a partial list of legislation pertinent to the financial services industry in 1.5.1, “Addressing regulatory requirements” on page 10.

1.2 What records management involves

Records management is a formal and structured process of identifying recorded information, of preserving needed content, and of destroying what is no longer needed after the approved retention period has been reached.

In simple terms, managing records requires the following actions:

- ▶ Categorizing records
- ▶ Retaining records for a specified length of time
- ▶ Destroying records when the company is no longer obliged to retain them
- ▶ Retaining an audit trail of all activity

There are two key factors in records management:

Preservation	Make sure to keep only what you need to keep for as long as you need to keep it.
Destruction	Make sure that after the required retention period ends, records are destroyed.

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

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

Not all content is a *record*. Ideally, you are able to automate the identification of this subset and manage it formally under a records management program. All other content, as part of a modern information governance program, should also be under some form of simple retention control to avoid the historical habit of just

keeping it all indefinitely. The preferred practice is to define and apply an automated general retention period across all relevant content and, under a records management program, automate the identification, declaration, and disposal of the subset that must be managed differently,

An effective records management solution manages the lifecycle of corporate records from creation to cremation. Each record has its own *lifecycle*. In its inception, contents are created or captured. The content is then organized, used, and disseminated. At some point, the contents of the records are declared as *records*. Records are preserved and retained. At the end of their lifecycles, records are disposed of as specified.

A record is only one type of content that falls into the domain of content management. At the inception of a record, an author creates a document. There can be many revisions to this document. When the document becomes an official record, it cannot be altered and is now subject to a *retention rule*. When it is time for disposition, an authorized person can archive or expunge the record. Typically, the disposition process is automated.

Note: *Expunge* is a records management term that implies irrevocably deleting the records so that even document forensics cannot recover any aspect of the records. Records are expunged when *destroying* is a *records disposition* option.

Records management is about retaining corporate records for the appropriate retention period to meet the business and regulatory requirements. The essence of records management is managing the risks and costs of retaining corporate records. Companies are required to demonstrate that they have records retention policy and procedures in place and that they enforce these policy and procedures consistently.

1.3 The business challenge: Information lifecycle governance

In earlier days when there were only physical records—predominately paper record and microfiche—staff used to file hard copies of final versions of documents according to a company's retention policies. Drafts were discarded. Index cards were used for cataloging the documents. When the retention period expired, records were disposed of. In some cases, records were kept permanently.

As technology has advanced with the invention of analog fax, printers, and wide use of computers, the volume of records has increased. Before electronic records had legal effects, companies still managed official records by keeping physical records in a somewhat controlled environment.

In today's digital world, records can be in the form of paper records, microfiche, electronic documents, electronic mail, fax, instant messages, collaboration content, voice recordings, wireless communication content, audio, video, shared drive content, and Web content. Electronic documents are now valid and have legal effect. They can be subpoenaed by opposing counsel. All of these contents are subject to legal discovery and can be produced as evidence if there is litigation and for the purpose of potential of records for audit, US Freedom of Information Act (FOIA) request, regulatory investigations, and so on.

In an extreme case where records are being kept indefinitely, they could be detrimental to the company if there is litigation, because the records are sometimes not to the company's advantage.

Companies are facing information that grows exponentially and spans various media, such as email, shared drives, packaged products, or cloud storage. This puts pressure on existing company infrastructure, drives up costs, and lowers efficiency.

From the compliance perspective, companies are dealing with discovery risks related to meeting legal obligations to preserve and provide evidence in case there is litigation or an investigation. The cost of meeting the legal obligation is high, and not being able to meet the obligation can damage the company's reputation and good will.

C-level executives who are dealing with explosive growth of information must decide what to retain and what can proceed to disposition. This leads to unnecessary IT costs, which takes away from strategic initiatives of the companies.

The number of documents and records that business needs to manage has increased exponentially by the use of other communication media. Although most of the companies have a good process for managing physical records, they need to extend the records management program to cover records produced by the other media, such as electronic documents and email.

1.4 The importance of records management

In a corporate environment, documents are often created or captured in a decentralized environment with no surveillance. Documents are named and filed according to the individual's preferences and often duplicated. Records are kept for too long, which can lead to increased storage cost. If the records become part of litigation, companies spend more to locate the records. In some cases, companies cannot locate the records, which can lead to a financial penalty or, more importantly, damage to the company's reputation.

The key objectives of records management include risk mitigation and cost containment of recordkeeping. There are several benefits:

- ▶ *Operational efficiency.* Making sure that corporate information is captured, retained, and disposed properly (which is one of the keys to an efficient company).
- ▶ *Cost containment.* Making sure that records are destroyed after their required retention period can reduce storage costs and space requirements.
- ▶ *Meet compliance and litigation requirements.* Industries and government regulations often impose different retention requirements for records. Timely destruction of records in full compliance reduces the risk of exposure in case of litigation.
- ▶ *Safeguard records for business continuity audit or business continuity reasons.* Records are vulnerable to natural disasters, accidents, theft, or mishandling. An efficient records management solution helps identify and protect against such threats, which is especially important for vital records that are essential to the continuation of the operation of a company.
- ▶ *Compliance with fiscal requirements.* Effective management provides verification that any fiscal constraints on records keeping are met.
- ▶ *Keeping a history of records.* An active, well-defined and documented records management system helps in defending why records were deleted and are no longer available.

Businesses need a holistic approach to record management throughout a record's lifecycle, from capture, through retrieval and archiving, to 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 verifies integrity and authenticity to prove compliance, especially during an audit.

1.5 Legal, regulations, compliance, and investigations

The legal, regulations, compliance, and investigations domain addresses general records retention legislation and regulations.

Figure 1-2 attempts to provide a glimpse of the global legal landscape pertaining to records retention. A company might be subject to jurisdiction in more than one country, depending on where the company is incorporated and files taxes, where the company is located, and whether the employees are within that country.

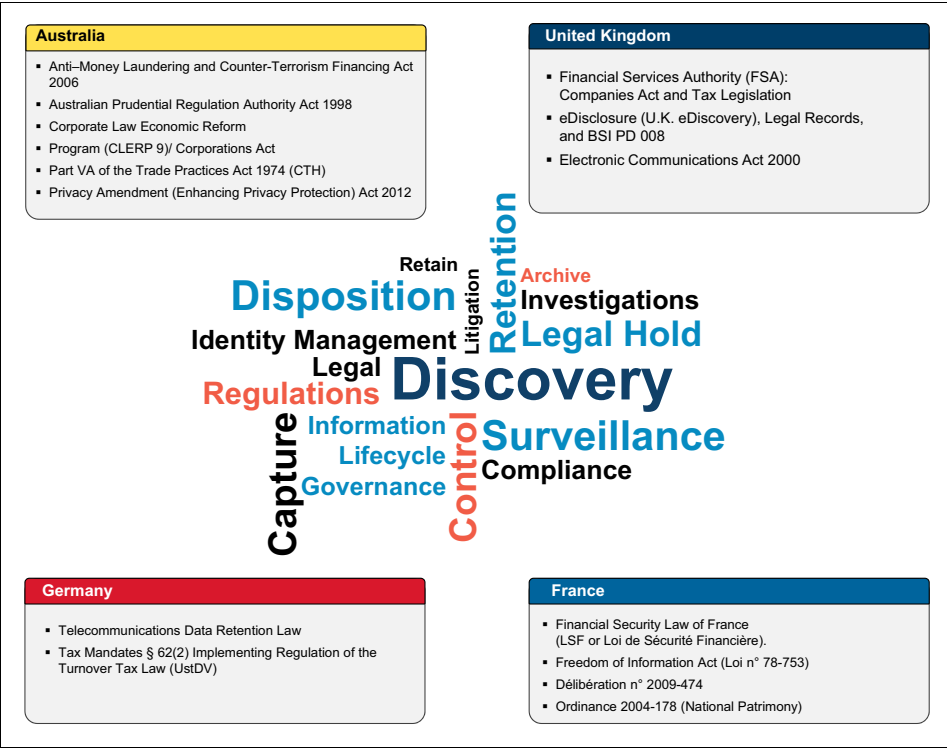


Figure 1-2 Regulations in different countries

Note: This section is about the conceptual issues and concerns and is not intended as a deep technical discussion of the domain. The focus of this chapter is on concepts and internationally accepted methods, processes, and procedures. This chapter avoids in-depth descriptions of country or region-specific laws, legislation, and regulations. Although some regional examples are presented to clarify certain points, these are limited to the emphasis of principles common across most jurisdictions, if not all of them.

In this section, we provide certain major records retention legal regulations that pertain to financial institutions in the United States (US) and laws that relate to information systems. The intention is not to turn readers into international law experts but to introduce the context and backdrop for the remainder of the chapter. We also examine the need for awareness of legislative and regulatory compliance. This includes general information system legislative and regulatory principles. We then move to investigations.

Regulatory bodies and the government impose different retention periods on different *record series*.

Note: A *record series* is a group of related records grouped as a unit and evaluated as a unit for retention and disposition purposes.

Compliance specifies how and what documents a company needs to preserve to comply by laws and regulations. One of the objectives of an effective records management program is to preserve records for the appropriate length of time. Companies that destroy records before their legal retention period expires can be subject to adverse consequences if there is litigation. Alternatively, exceeding the required retention period can put the company at a disadvantage in litigation, and it also leads to higher storage and discovery costs.

Whenever there is a court or regulation authority order, companies must go through a legal *discovery* process. Often, this requires the companies to search all documents to determine whether they are *records* or not and identify those that match the discovery order. Any document in any medium that has information relevant to the subject matter of a dispute is potentially *discoverable* and must be preserved for as long as the lawsuit is anticipated, pending or in process. These records and documents need to be placed on *record hold* so that the normal retention schedule and disposition is no longer applicable during the process.

Note: IBM Enterprise Records has *record hold* capabilities. You can apply record holds only to content that is declared and managed as records. In general, this is guided by eDiscovery for litigation.

A *record hold* is different from a *legal hold*, which is an action taken on record collections to ensure that they are not disposed of as part of their normal retention schedule lifespan and are kept, possibly 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 guided by litigation discovery needs.

Legal holds are commonly scoped and apply across all relevant content in the business. The fact that you defined some of it as “records” is immaterial.

During discovery, in some cases documents are produced out of context, are damaged, or are presented to the court and can damage the company’s case.

Companies might not be able to claim *undue burden* as a reason of not being able to produce relevant records in response to discovery orders. The inability of a company to comply with a regulation or legal action can result in financial loss or damage to the company’s reputation.

1.5.1 Addressing regulatory requirements

Companies need to have a good understanding of applicable regulations, identifying records, and identifying the corresponding retention requirements for the records. Complexity of managing records is increased by evolving compliance rules and regulations, as illustrated in Figure 1-3 on page 11.

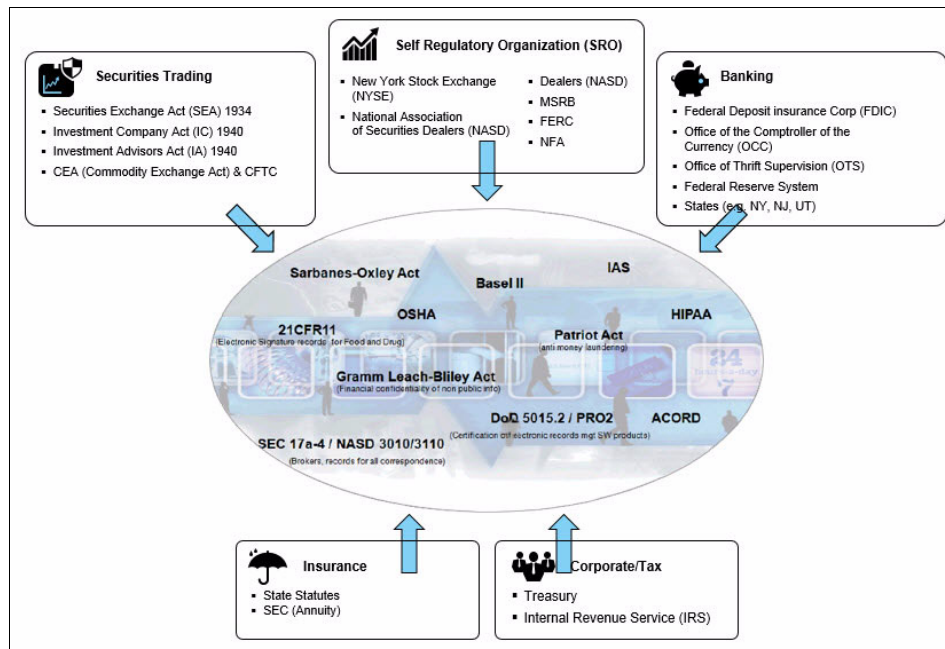


Figure 1-3 Example of regulations pertaining to retention in the US

Compliance is the act of adhering to and demonstrating adherence to internal or external regulations. A *regulation* 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 transactions in the financial sector.

Compliance with what any regulation and law requires the following actions:

- ▶ Interpreting what the regulation and law says
- ▶ Verifying 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 straight-forward exercise because of the complexity of the regulations.

Examples of US regulations related to financial transactions

This section lists selected regulations that pertain to the global financial institution in our case study. They provide an example of how different regulations have different requirements across the different legal entities. Chapter 3, “Retention and file plans” on page 65, describes the case study and demonstrates how the legal requirements are analyzed and apply to the retention schedule.

Securities traders

The selected US regulations that follow pertain to records of trading securities:

- ▶ **Securities Exchange Act (SEA) 1934**

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 electronic interoffice communications and communications with customers.

Sec. 240.17a-1 Recordkeeping rule for national securities exchanges, national securities associations, registered clearing agencies, and the Municipal Securities Rulemaking Board.

Sec. 240.17a-2 Recordkeeping requirements relating to stabilizing activities.

Sec. 240.17a-3 Records to be made by certain exchange members, brokers, and dealers.

Note: 17a-3 requires that all members of a national securities exchange, including all brokers and dealers, keep current a variety of books and records that relate to their businesses.

Sec. 240.17a-4 Records to be preserved by certain exchange members, brokers, and dealers.

Note: 17a-4 requires that some records that must be 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.

Note: 17a-4 requires broker-dealers to maintain records electronically by using a digital storage medium that preserves the records exclusively in a *nonrewriteable, non-erasable* format.

Sec. 240.17f-1 Requirements for reporting and inquiry with respect to missing, lost, counterfeit, or stolen securities.

Sec. 240.17f-2 Fingerprinting of securities industry personnel.

Sec. 240.17h-1T Risk assessment recordkeeping requirements for persons associated with brokers and dealers.

Sec. 240.17Ad-6 Recordkeeping.

Sec. 240.17Ad-7 Record retention.

Sec. 240.17Ad-11 Reports regarding aged record differences, buy-ins and failure to post certificate detail to master security holder and subsidiary files.

Sec. 240.17Ad-15 Signature guarantees.

► Investment Company Act (IC) 1940

Sec. 270.31a-1 Records to be maintained by registered investment companies, certain majority-owned subsidiaries thereof, and other persons having transactions with registered investment companies.

Sec. 270.31a-2 Records to be preserved by registered investment companies, certain majority-owned subsidiaries thereof, and other persons having transactions with registered investment companies.

Sec. 270.31a-3 Records prepared or maintained by other than the person required to maintain and preserve them.

Sec. 270.38a-1 Compliance procedures and practices of certain investment companies.

► Investment Advisors Act 1940

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

Sec. 275.204-2 Books and records to be maintained by investment advisers.

► CEA (Commodity Exchange Act) and CFTC

Sec. 1.31 Books and records, keeping and inspection

Sec. 1.32 Segregated account, daily computation and record

Sec. 1.33 Monthly and confirmation statements

Sec. 1.34 Monthly record, point balance

Sec. 1.35 Records of cash commodity, futures, and option transactions

Sec. 1.36 Record of securities and property received from customers and option customers

Sec. 1.37 Customer's or option customer's name, address, and occupation recorded; record of guarantor or controller of account

Sec. 1.39 Simultaneous buying and selling orders of different principals;
execution of, for, and between principals

Sec. 42.2 Compliance with Bank Secrecy Act

Banks

The selected following are pertain to banking:

- ▶ Federal Deposit insurance Corp (FDIC)
 - Sec. 9.8 Recordkeeping
 - Sec. 27.3 Recordkeeping requirements
 - Sec. 27.5 Record retention period
- ▶ Office of Thrift Supervision (OTS)
 - Sec. 551.50 What records must be maintained for securities transactions
 - Sec. 551.60 How records must be maintained

Self-regulatory organizations

The selections that follow pertain to self-regulatory organizations (SROs):

- ▶ National Association of Securities Dealers (NASD)

Note: NASD rules 2711, 3010 and 3110 requires that member firms establish and maintain a system to supervise the activities of each registered representative, including transactions and correspondence with the public.

Rule 2210 Communications with the Public

Rule 3010 Supervision

Rule 3011 Anti-Money Laundering Compliance Program

Rule 3060 Influencing or Rewarding Employees of Others

Rule 3110 Books and Records

Rule IM-3110 Customer Account Information

Rule 3115 3115. Requirements for Alternative Trading Systems to Record and Transmit Order and Execution Information for Security Futures

- ▶ 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.

Related regulations

The following list includes some of the other regulations:

- ▶ **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.¹
- ▶ **Statutes**
State or local laws also govern the requirement of recordkeeping. Each state has its own jurisdiction, depending upon the state where you are located.
- ▶ **Sarbanes-Oxley Act (SOX)**
The Sarbanes-Oxley Act requires firms that audit companies governed by the SEC to retain all relevant documentation to protect against mishandling of information.
- ▶ **Gramm-Leach-Bliley Act**
The Gramm-Leach-Bliley Act (GLBA), also known as the Financial Services Modernization Act of 1999, is a US federal law enacted to control ways that financial institutions to deal with private information for individuals.
- ▶ **The Office of Foreign Assets Control**
The Office of Foreign Assets Control (OFAC) of the US Department of the Treasury administers and enforces economic and trade sanctions based on US foreign policy and national security goals against targeted foreign countries, terrorists, international narcotics traffickers, and those engaged in activities related to the proliferation of weapons of mass destruction.

1.5.2 Investigations

The requirements for the admissibility of evidence vary across legal systems and among different cases. At a generic level, evidence should have some probative value, be relevant to the case at hand, and meet the following criteria:

- ▶ **Authentic**
- ▶ **Accurate**
- ▶ **Complete**

¹ Information taken from the Internal Revenue Service website:
<http://www.irs.gov/businesses/small/article/0,,id=152532,00.html>

- ▶ Convincing
- ▶ Admissible

Two concepts are particularly important when dealing with digital or electronic evidence: *chain of custody* and *authenticity* or integrity.

The *chain of custody* refers to who, what, when, where, and how the evidence was handled from its identification through its entire lifecycle, which ends with destruction. Any break in this chain can cast doubt on the integrity of the evidence.

Ensuring the *authenticity* and integrity of evidence is crucial. If the courts feel that the evidence is not accurate or lacks integrity, it is doubtful that the evidence or any information derived from the evidence will be admissible.

Note: For example, paragraph (f)(2)(ii)(A) of SEC Rule 17a-4 requires that the electronic storage media used by broker-dealers preserve the records exclusively in a *non-rewritable* and *non-erasable* format.

1.6 Planning an information lifecycle governance program

Companies require a holistic *information lifecycle governance* (ILG) program to meet today's compliance and business needs. An impeccable plan illustrates the deep linkages between business and technology. As Figure 1-6 on page 19 shows, a well-informed ILG program considers the linkages between strategic business objectives, capabilities required to achieve those objectives, gaps in capability areas, initiatives required to close those gaps, and the prioritization of such initiatives.

As summarized in Figure 1-4 on page 17, business and IT stakeholders have different interests and motivations.

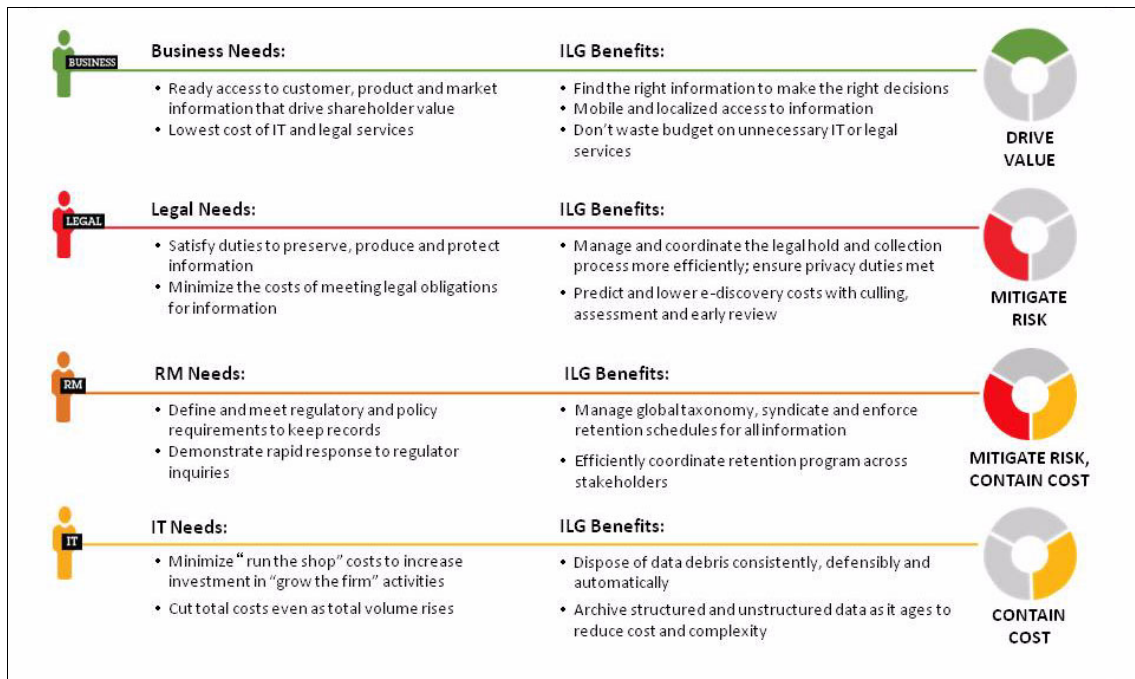


Figure 1-4 ILG needs and benefits from different stakeholders' perspectives

The business executives want to use information for better and more informed decision making. They would like to get mobile and localized access to information and would like to avoid unnecessary IT or legal costs.

Legal personnel want to meet their obligations to preserve, produce, and protect information. Their goal is to minimize the costs of meeting those legal obligations for information.

Records managers want to define and meet regulatory and policy requirements to keep records and to respond rapidly to regulator inquiries.

Information technology personnel want to minimize administrative costs and be able to channel investments and resources to strategic growth areas of the company. They want to achieve economies of scale to drive the costs down when volume rises.

Figure 1-5 summarizes the planning phases.



Figure 1-5 Planning an ILG program

1.6.1 Obtaining corporate sponsorship and stakeholder buy-in

A successful records management program requires corporate governance from the top down and enforcement throughout the company. Executive sponsorship is a key to the success of an enterprise-wide deployment. Other stakeholders include, but are not limited to, Records Managers, the 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 case of litigation. It is important to get an executive sponsor and to engage a cross-functional team for enterprise-wide participation.

1.7 Records management maturity model

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

1.7.1 Using an objective records management maturity model

One way to assess the health of a company's current records retention and management practices is to use an objective records management maturity model, from nonexistent through optimized stages, as shown in Figure 1-6.

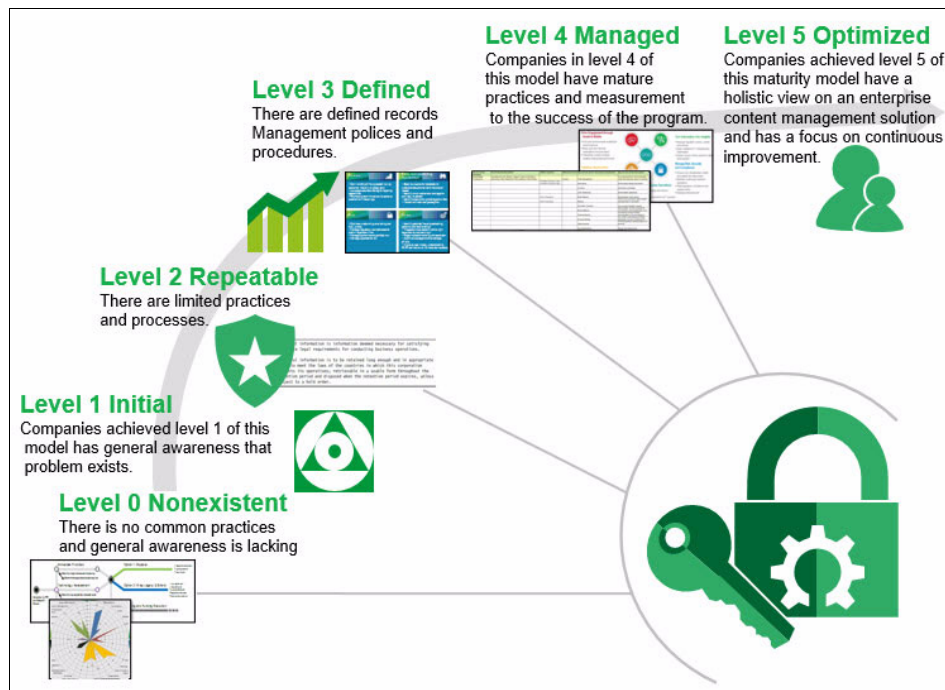


Figure 1-6 Maturity model

This model enables companies to establish baselines for business records retention and management practices by using an objective matrix:

- ▶ Level 0. Non-existent

There are no common practices and general awareness of records management within the company is lacking.

Companies at maturity Level 0 are likely to exhibit the following behaviors:

- No common practice on records retention
- No general awareness of records retention program
- Costly and manual legal discovery
- No formal archiving
- No information classification
- Inability to produce records when required

There is no system in place. Electronic discovery for companies at this level is typically an outsourced process.

► Level 1. Initial

Companies that have achieved Level 1 of this model have a general awareness that a problem exists. Currently, they have no common practices of records management policies and procedures in place.

Companies at maturity Level 1 might exhibit the following behaviors:

- Formal records program for physical records and, possibly, archiving
- Departmental policies exist but little or no awareness of them
- Some control over paper records
- User-controlled email and documents
- Information that originated in digital form 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 an email 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. Repeatable but intuitive

Companies that have achieved Level 2 have limited practices and processes.

Companies at maturity Level 2 might exhibit the following behaviors:

- Primitive records retention process that is repeatable
- Some awareness of enterprise needs on records retention
- Limited practices or records management tools

► Level 3. Defined process

Companies that have achieved Level 3 of this maturity model have defined records retention policies and procedures.

Companies at Level 3 might exhibit the following behaviors:

- Formal records program for archiving physical and electronic records
- Electronic discovery is still a costly, manual process
- Initial awareness of enterprise policies

- Some procedures for handling electronic records
- Manual declaration and classification by business users
- Electronic discovery partly supported by IT
- Key repositories for federation identified

Level 3 companies have an ERS system in place. They manage email and desktop files as records from the ERS. There are electronic discovery and collection tools and image capture is in place also. Classification is still manual. Storage and ERS are partially integrated.

► Level 4. Managed and measurable

Companies that have achieved Level 4 have mature practices. There are defined measurements of performance of the program.

Companies at maturity Level 4 might exhibit the following behaviors:

- Measurable records program
- Enterprise policies enforced and general awareness of the policies
- Increased control over electronic records
- Electronic discovery increasingly supported by IT
- Expanded paper conversion to reduce risk and cost

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

► Level 5. Optimized

Companies that have achieved Level 5 have a holistic view of an enterprise content management solution and has a focus on continuous improvement.

Companies at maturity level 5 are likely to exhibit the following behaviors:

- Records program embedded in key processes and the IT infrastructure
- Enterprise policies complete and enforced
- Integrated records management and electronic discovery processes
- Enterprise paper conversion programs in place
- Enterprise federation strategy in place

Companies that achieve this level of maturity have an ERS system that is 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.

1.8 Organizational readiness

Figure 1-7 is a sample visual representation of an output of the assessment that measures the readiness of an ILG within a company.

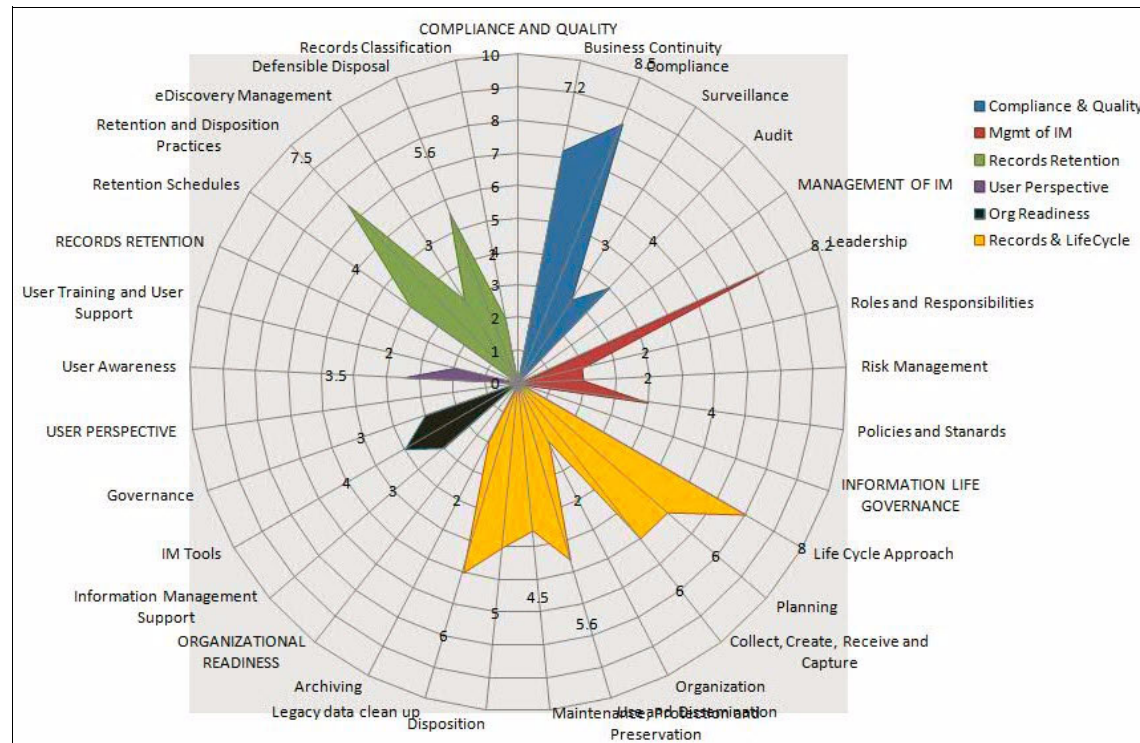


Figure 1-7 ILG assessment

The guidelines for this assessment include completing the following tasks:

1. Identify key measurement areas in the ILG program.
2. Identify departments that receive records management services. Conduct interviews with selected departments.
3. Review the results from the interviews and address specific issues or concerns raised by the departments.
4. Analyze the results of the survey and determine the departmental satisfaction rating for each key measurement area.
5. Compare the departmental satisfaction rating with the target. Determine the course of action needed for improvement, and track those actions.

This type of visual representation, called a *radar* graph (also called a *star* or *spider* graph), can serve as a guideline for assessing 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 result.

The next step is to identify the gaps and to determine the business and technology requirements.

Each competency can be further broken down into ILG processes. For example, the following are sample processes related to legal requirements:

- ▶ Identifying employees
Determining which employees have information potentially relevant to an actual or anticipated lawsuit or government investigation.
- ▶ Identifying data
Determining information, records, and data sources that are potentially relevant to an actual or anticipated lawsuit or government investigation.
- ▶ Notification of the hold
Communicating, syndicating, and executing legal holds to people, systems, and data sources for compliance.
- ▶ Evidence collection
Fact-finding and inquiry with employees who have knowledge of a matter in dispute to determine potentially relevant information, the sources, and their locations. Collecting potential evidence in response to a request agreed to with an adversary or government agency.
- ▶ Evidence analysis and cost controls
Assessing information to understand dispute and potential information sources and for determining, controlling, and communicating the costs of outside review of relevant information.
- ▶ Legal record
Documenting the custodians and data sources identified and the legal hold and collection activities over multiyear lifecycle of the material.

This example pertains to records management:

- ▶ Master retention schedule and taxonomy
Defining an information classification schema that reflects the organization structure; cataloging, updating, and mapping the laws that apply to each class in the countries where the organization operates to determine regulatory

record-keeping obligations; establishing and managing a network of records liaisons to help establish what records exist and where.

The following examples pertain to the business:

- ▶ Departmental information practices
Using an enterprise information taxonomy, cataloging which information each business organization values, generates, or stores by class, where they store it, and how long it is useful to them results in retention schedules for information and enables data source-specific retention schedules that reflect both business value and regulatory requirements.
- ▶ Realize information value
Gaining timely access to and ability to apply information in the course of their work, including the ability to harness information of quality as it ages and the ability to use relevant information with or without author context to maximize the enterprise value of information.

The following are examples pertain to privacy:

- ▶ Secure information of value
Determining a schema for the various levels of information importance and the corresponding security needed; using an enterprise information taxonomy and network of liaisons across the business, cataloging which information each business organization generates or stores and assigning the appropriate security level; communicating these security needs to employees who generate, use, manage, and store information.
- ▶ Privacy and data protection
Assessing privacy duties by data subject and data location, including overlapping obligations for information and information elements and a means of communicating these requirements to those employees who generate, use, access, and store information.

The following are examples that pertain to information technology:

- ▶ Data source catalog and stewardship
Establishing a common definition and object model for information and the people and systems with custody of it for use in determining, defining, communicating, understanding, and executing governance procedures
- ▶ System provisioning
Provisioning new servers and applications, including associated storage, with capabilities for systematically placing holds, enforcing retention schedules, disposing, collecting evidence, and protecting data elements subject to privacy rights

- ▶ Active data management
Differentiating high value actively used data by the business from aging data of value to regulators only or less frequently accessed data; results in increased accessibility, security, privacy; aligns and enables data value with storage tiering by value
- ▶ Disposal and decommissioning
Disposing of data and fully decommissioning applications at the end of their business utility and after legal duties have elapsed
- ▶ Legacy data management
Processes, technology, and methods by which data is disposed of and applications are fully decommissioned at the end of their utility and after legal duties have elapsed
- ▶ Storage alignment
The process of determining and aligning storage capacity and allocation with information business value and retention requirements, including optimizing use targets, storage reclamation, and reallocation after data is deleted to link storage cost to business need for data stored
- ▶ Audit
Testing to assess the effectiveness of other processes, which, in this instance, refers to the processes for determining, communicating, and executing processes and procedures for managing information based on its value and legal duties and disposing of unnecessary data

1.9 Records management system technical standards and guidelines

A *records management application* is a software tool to help solve a business need that often involves process changes. Standards for records management are emerging and evolving in many countries around the world.

Records management standards are the guidelines to manage records as 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.

Note: *Guidelines* are recommendations or *non-mandatory* controls that help to support standards or serve as references when no applicable standards are in place.

These product standards provide a *baseline* for the technical requirements. What follows are examples of some of the records management standards around the world:

- 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 enables an organization to efficiently and effectively record and retrieve information, which enhances decision-making, productivity, and accountability and reduces 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 Electronic Records Management Systems (ERMS), this implementation is considered an enabler for ISO 15489.²

- US 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 US 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.

IBM Enterprise Records is DoD certified since inception. DoD 5015.2 is also the starting point for base use cases for the retired UK National Archive (TNA) and the new European MoReq2010 standards.

For example, one of the mandatory requirements is C2.2.4.1: Records management applications (RMAs) shall treat email messages as any other records, and these shall be subjected to all requirements of this standard.

These standards are evolving. For example, C2.2.4.5 of version 3 of DoD 5015.2 mandates that RMAs shall not require users to save attachments to

² ISO official website: <http://www.iso.org/iso/home.htm>

their hard disk drives or other media before filing them separately from the email message. This is new to version 2.

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

A third revision of the specification came out in 2007. Version 3 added these provisions:

- 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* means to transfer and archive records from one records management system to another records-holding authority. It is one type of record disposition option.

► Model Requirements for Managing Electronic Records (MoReq 2010)

The MoReq specification is a model specification of requirements for ERMS to be used in Europe. For example, 5.1.1 of the specification stipulates that the ERMS must provide a function that specifies retention schedules, automates reporting and destruction actions, and provides integrated facilities for exporting records and metadata.

MoReq 2010 is the next generation of the MoReq standard. MoReq 2010 was formally published June 2011. The former version MoReq2 was 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 and provides a software certification testing program for vendors. It is a European standard, but different countries can have local variations. MoReq 2010 was written to encourage different models of records management system to emerge.

► Document Management and Electronic Archiving (DOMEA)

DOMEA is a German standard for document management and electronic archiving in public administration. For example, requirement group (RG) 5 stipulates requirements about mobile records management. In addition to records management through web clients, mobile records management represents an alternative for many authorities to ensure the fulfillment of daily tasks, regardless of the employees' presence in the office.

► Victorian Electronic Records Standard (VERS)

VERS is an Australian standard developed by Public Record Office Victoria (PROV) to provide guidelines on capturing, managing and preserving electronic records in the state of Victoria.³ For example, it defines that an electronic records format must be able to support evidence. Electronic records must be admissible as evidence and given due weight in a court of law. This requires the ability to prove that a record has not been altered in an unauthorized or undocumented fashion since creation and to demonstrate who created the record and when it was created.

Although the other standards mentioned here really focus on requirements for a records management solution, VERS concentrates on defining a standard for the long-term preservation of electronic records. The intention is to ensure that an electronic record created today using current technology can fulfill these objectives:

- 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.
- Have context, so that it is understood exactly where the record came from, who the author was, and what it is related to.

1.10 Role of IBM Enterprise Records within the IBM Information Lifecycle Governance portfolio

The IBM Information Lifecycle Governance (ILG) portfolio includes capabilities that legal, IT, records, and business users can use to help manage legal risk and to reduce data management and discovery costs. To achieve this, the ILG portfolio provides solutions for electronic discovery (eDiscovery), archiving, disposal, policy management, and records retention and management. The portfolio helps organizations manage enterprise information based on its business value, comply more efficiently with litigation and regulatory duties, and dispose of information when it is no longer required. Figure 1-8 on page 29 provides a high-level overview of the ILG portfolio.

³ According to the Public Record Office of Victoria, Australia
<http://www.prov.vic.gov.au/vers/standard/>

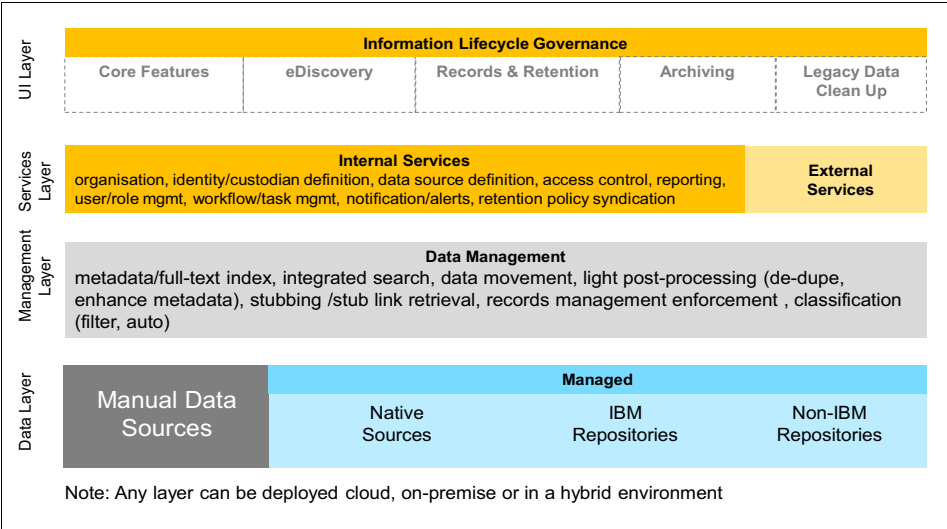


Figure 1-8 IBM Information Lifecycle Governance portfolio

In terms of the broader ILG portfolio, IBM Enterprise Records is incorporated in the Records Retention and Management element of the portfolio. Records Retention and Management helps you define and describe the records retention policies for your organization. Those policies can then be applied to content managed in an IBM repository by using IBM Enterprise Records.

IBM Enterprise Records is an electronic records management application to help manage ongoing governance for records, from creation to disposition, to ensure that records remain trusted, accurate, and compliant through the application of defensible and relevant records retention policies. Enterprise Records combines content, process, content federation technology, and connectivity to automate and simplify all record-based activities.

1.10.1 Policy management relationship to IBM Enterprise Records

In today's business environment, many organizations face several records retention and management challenges as a result of their standard operations:

- ▶ Overlapping or conflicting retention schedules that specify the duration that information must be retained
- ▶ Requirements to apply multiple schedules to information that is specific to the origins of the information, for example, retention policies specific to Europe, Singapore, and New Zealand

- ▶ Difficulties as a result of the size and structure of the organization, such as multiple departments that need to add the same record type, which is separately securable or needs to be reviewed by different business areas
- ▶ Requirements to apply policies to physical and electronic data and to manage items existing in both formats, individually or in unison

As an initial step, many organizations begin recording the various regulatory obligations for their enterprises in tools such as spreadsheets, which often show no relationship or bear little resemblance to the application that they use for records management. To assist with overcoming these complexities, IBM Global Retention Policy and Schedule Management policy and the Retention Management module offer several capabilities:

- ▶ Define and capture accurate, dynamic schedules that are value-based and support the distinct needs of different business units and countries
- ▶ Host a full, shared law library to create a centralized repository to support legal requirements
- ▶ Establish information lifecycles and enable auditable programs for both physical and electronic data
- ▶ Synchronize and maintain centralized retention control to deliver a consistent corporate retention framework with local responsibilities

After the organization's regulatory obligations are defined and captured in Global Retention Policy and Schedule Management, they are available to be syndicated to the organization's unstructured and structured data management applications, such as IBM Enterprise Records for unstructured data management or IBM Optim™ for structure data management. The syndication of data from Global Retention Policy and Schedule Management provides Enterprise Records with the structure for managing the organization's records, including the file plan, the retention schedules, the disposal schedules and disposal actions, and the record information owners.

Figure 1-9 on page 31 provides a conceptual overview of Global Retention Policy and Schedule Management. It illustrates how the software creates policies based on the organization, what information is stored, where, why, and when to preserve and retain information. It is the unification of this information that can be disseminated to other systems by using Global Retention Policy and Schedule Management policy syndication.

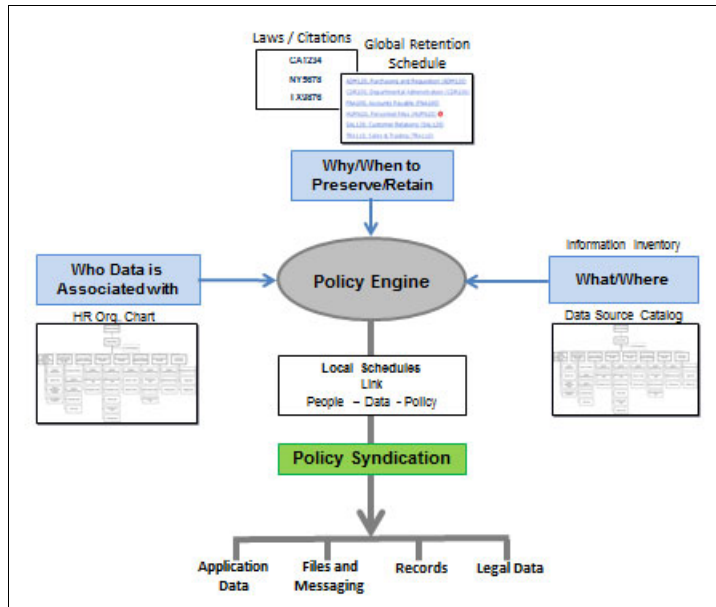


Figure 1-9 Overview of Global Retention Policy and Schedule Management Policy Management

It is also possible to manually enter or import this information into Enterprise Records even if you do not have Global Retention Policy and Schedule Management. Many IBM clients also use custom scripts to import the file plan and retention rules.

1.10.2 Syndicating global retention and schedule management policies to Enterprise Records

Where an organization has used Global Retention Policy and Schedule Management to define the organization's corporate and classification structures, there is a mechanism to syndicate the details to Enterprise Records.

Figure 1-10 on page 32 describes, at a high level, the information from Global Retention Policy and Schedule Management being syndicated to Enterprise Records to create the structure of the file plan.

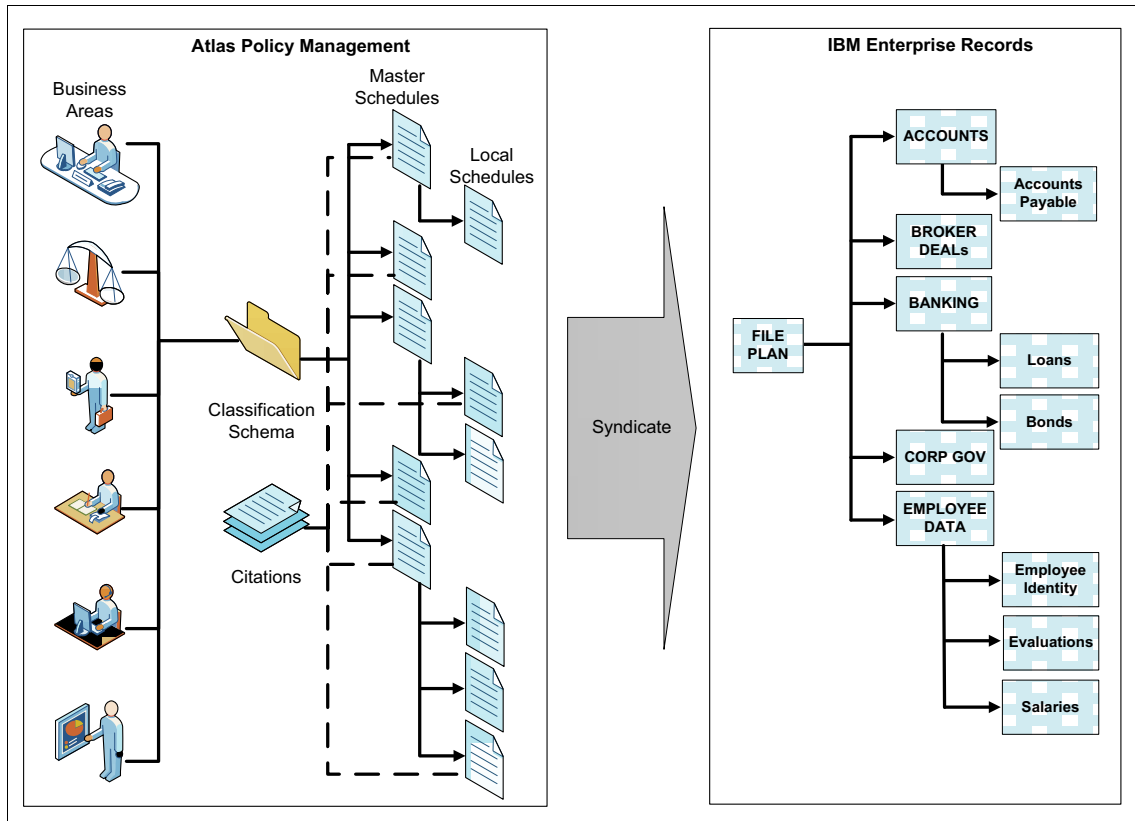


Figure 1-10 The relationship between Global Retention Policy and Schedule Management Policy Management and IBM Enterprise Records

The left side of the diagram shows IBM Global Retention Policy and Schedule Management, which is used to create the Records Retention and Management policy for the organization, based on organization structure, the corporate taxonomy, and library of retention, privacy, and discovery laws with appropriate roles and change management processes for field-level authorization. Using configuration settings, items identified for use in Enterprise Records can be flagged for syndication.

The right side of the diagram shows IBM Enterprise Records, which has the file plan and disposal authorities. These elements are populated directly from Global Retention Policy and Schedule Management by using the IBM Atlas eDiscovery Policy Syndication Framework.

Important: When Global Retention Policy and Schedule Management is used to create or define the structure in IBM Enterprise Records, the intent is for Global Retention Policy and Schedule Management to continue to be used as the Records Retention and Management policy tool for the organization. All changes or modifications to the schedules and policies should be enacted in Global Retention Policy and Schedule Management, which is then syndicated as an update to IBM Enterprise Records.



IBM Enterprise Records system and architecture

IBM Enterprise Records software can help an organization automate all record-based activities by supporting the entire lifecycle of records, from creation to disposition. This chapter describes the system and architecture and covers the following topics:

- ▶ Overview of IBM Enterprise Records software
- ▶ System architecture
- ▶ Data model, workflow, and security
- ▶ Logging
- ▶ User and administrative applications
- ▶ APIs and the Component Integrator
- ▶ Reporting
- ▶ References

2.1 Overview of IBM Enterprise Records

IBM Enterprise Records is an electronic records management application. It provides ongoing governance for records to help ensure that they remain trusted, accurate, and compliant with the enterprise's records retention and management policies. This is done through defined records retention policies that are defensible and relevant. Enterprise Records combines content, processes, content federation technology, and connectivity to automate all record-based activities by supporting the entire lifecycle of records from creation to disposition.

By supporting full record declaration automation, the solution ensures best practices for capturing records, categorization, and records administration while minimizing the impact on the business users and Records Managers.

Enterprise Records also gives you the flexibility to create single or multiple file plans for 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, typically containers. These might be organized to segment data according to retention policies and management considerations or for separation for access and ownership reasons.

2.1.1 Key business benefits of IBM Enterprise Records

Enterprise Records applies the retention schedule to records that have been declared. It automates many of the records management functions for electronic records to relieve the burden of managing the retention schedule from the records management team and other employees. Enterprise Records can capture, declare, categorize, store, secure, audit, report on, and dispose of electronic and physical records according to fiscal, legal, and regulatory requirements. This helps organizations handle several important activities:

- ▶ Comply with regulations by providing a secure, central repository for records. This repository automates the enforcement of records management policies and automates review of records and disposition of records.
- ▶ Respond promptly to regulatory inquiries or internal audits by including pre-built records management-specific report templates and custom reports. This can include the use of IBM Cognos® BI for report generation.
- ▶ Implement the records management policy for content in other enterprise content management (ECM) repositories, such as OpenText and Documentum, by using content federation technology to secure the records in-place and manage their lifecycles.
- ▶ Automate the capture and categorization of records by providing options to automate the collection and auto-classification of documents in bulk.

- ▶ Enterprise Records includes tools, at no charge, that automatically archive records from email systems, file shares, and Microsoft SharePoint.
- ▶ IBM StoredIQ® includes tools at no charge to analyze content. StoredIQ provides scalable analysis and governance of unstructured data in place across disparate and distributed email, file shares, desktops, and collaboration systems. StoredIQ enables companies to discover, analyze and act on data for eDiscovery, records retention and disposition, compliance, and storage optimization initiatives.

2.1.2 Software highlights and capabilities

IBM Enterprise Records is a records management application that offers the following standard records management functions:

- ▶ 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

However, Enterprise Records is more than just a records management application. Because of its integration with IBM FileNet Content Manager, the following benefits and capabilities make it greater than the sum of its parts:

- ▶ Fully integrated architecture and repository
- ▶ Intuitive user interface
- ▶ Comprehensive search
- ▶ Federated records management
- ▶ Compliance infrastructure (not a point solution)
- ▶ Automated records capture
- ▶ Active compliance

Enterprise Records combines content, workflow, and compliance services into an infrastructure that is built upon a single, fully integrated repository that manages all of your content (documents, email, and records). Built upon the latest platform infrastructure, the Enterprise Records user interface has been reimplemented using IBM Content Navigator technology, which provides significant improvements for user efficiency and reduces the number of clicks for Records Managers for file plan management, holds management, reports, and disposition reviews. Enterprise Records can perform consolidated searches and retrievals for records by content and metadata. 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 event-based architecture of FileNet

Content Manager that enables active content in a workflow, Enterprise Records creates an active compliance infrastructure that uses automated records capture technology.

2.1.3 Working with IBM Enterprise Records

Content Navigator is the unified experience for mobile, web, and desktop access across the IBM Enterprise Content Management portfolio. The Enterprise Records user interface is implemented by using Content Navigator technology. Figure 2-1 illustrates the Enterprise Records user interface for records management system administrators, Records Managers, and Privileged Users.

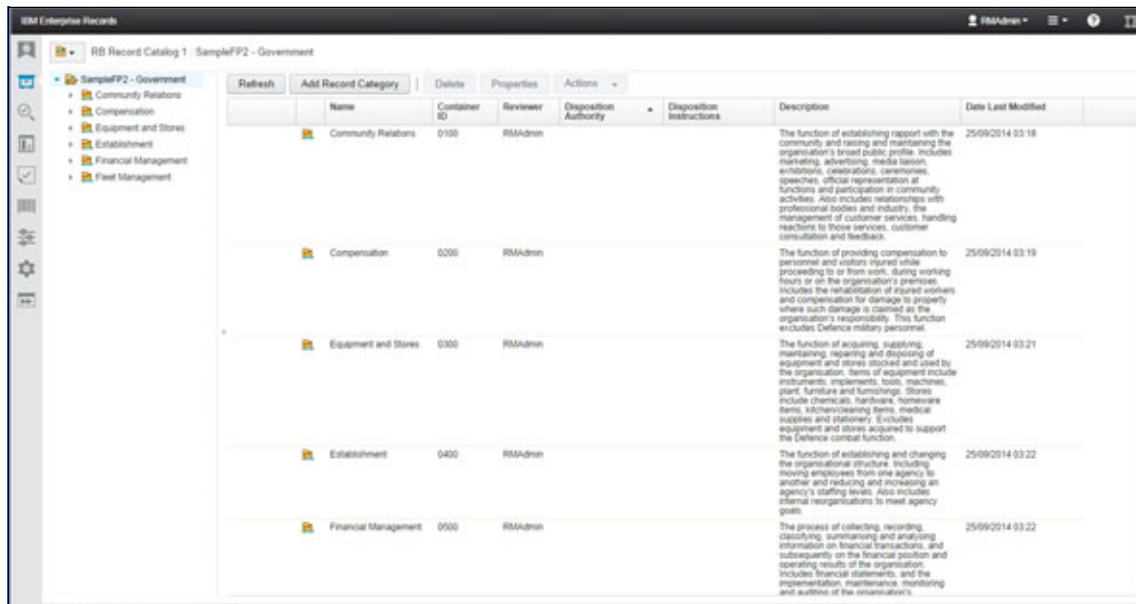


Figure 2-1 Screen capture of IBM Enterprise Records administrative client

Functions

The following records management functions are available for authorized records management users within this user interface:

- ▶ Browse, create, modify, and update file plans
- ▶ Search for records management assets and create new records management searches
- ▶ Create and manage records management entry templates
- ▶ Generate and review records management reports

- ▶ Review, run, and schedule records management tasks, such as sweeps
- ▶ Create, manage, and work with physical items, including bar codes and locations
- ▶ Create, manage, and work with record holds
- ▶ Manage and work with records management workflow

Authorized users

Authorized users are roles with various privileges for performing certain record management tasks.

Records Administrator

Enterprise Records treats the Records Administrator as the Enterprise Records system administrator. It is this user who accesses the Enterprise Records interface and configures access to desktops and repositories for the organization's Enterprise Records users. The Records Administrator completes this configuration by using the Open Administration view on the desktop.

We suggest that the Enterprise Records system administrator work with the Records Manager to define and schedule sweeps, because they will affect performance and should not be run concurrently with system backups.

Records Manager

Within Enterprise Records, the Records Manager is primarily responsible for monitoring the records in the system, placing records on hold, initiating disposition, running and scheduling records management reports, managing physical records and their movements, and approving records management workflows.

For organizations that are not using IBM Global Retention Policy and Schedule Management, the Records Manager works with the enterprise's records management professionals to build and maintain the file plan and all of its related elements and to make any adjustments to the file plan and disposition schedules as business or regulatory requirements change.

Privileged user

Enterprise Records can be used to provision access to some records management features of the Enterprise Records application through the configuration of a desktop that users log in to by using their standard logon credentials. This meets the needs of organizations that require *privileged users*, or users who fulfil a particular role, such as the departmental Information Coordinator, to have access to some records management features with minimal configuration. An example of a feature available to the Information Coordinator

would be ability to create and manage Enterprise Records entry templates for users in a department needing to manually declare items.

Creating, capturing, and declaring records

In an enterprise, content can come from a variety of sources, including scanned images, faxes, email, electronic documents, eForms, and web content. It is imperative that any enterprise content management (ECM) platform provides mechanisms for easily ingesting content from these disparate content sources and manage them as records in a file plan. IBM Enterprise Content Management provides a broad set of capabilities and integrated applications to ingest content from these sources and automatically declare and classifying them as records in the file plan. These products typically target a particular content source:

- ▶ IBM Datacap: Works with scanned images and faxes.
- ▶ IBM Content Collector: Works to collect the following documents:
 - email from email servers
 - Electronic documents on network file shares
 - Electronic documents stored in Microsoft SharePoint
 - Electronic documents stored in IBM Connections
- ▶ IBM Case Foundation and IBM Case Manager: Works with documents attached to cases in case workflows
- ▶ IBM Content Classification: Works with documents stored in FileNet Content Manager.
- ▶ IBM Content Navigator: 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 software suite.
- ▶ Entry Templates: Enable an administrator to customize the declaration wizards providing default values, automating data collection, and minimizing user input.

Figure 2-2 on page 41 shows how these products provide sources for capturing business records in Enterprise Records.

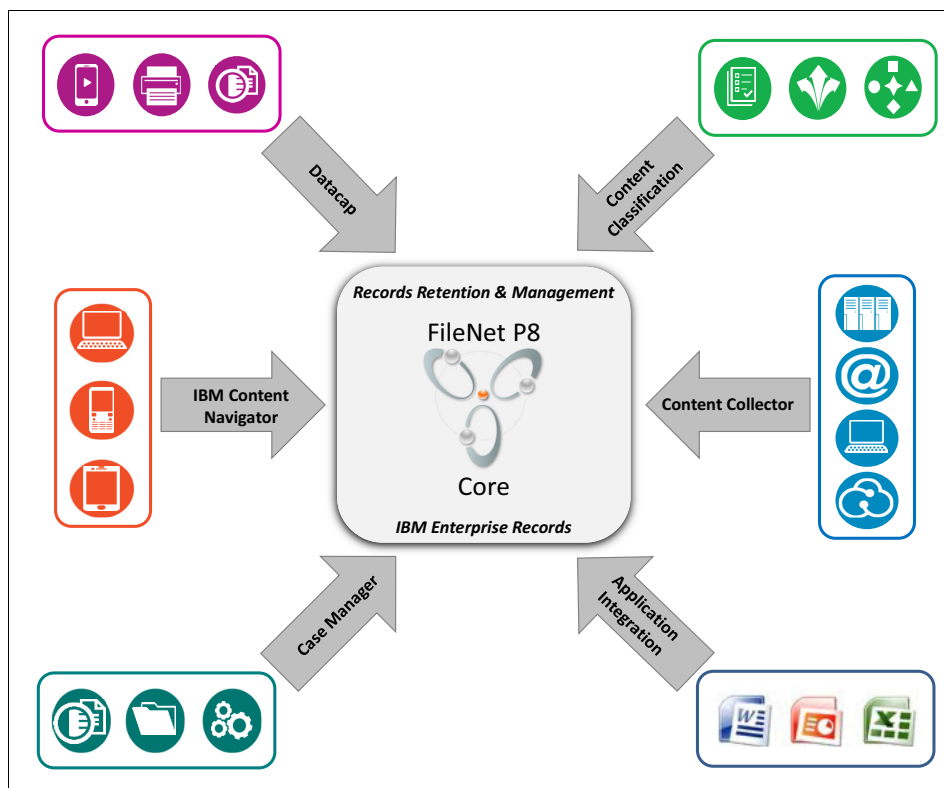


Figure 2-2 IBM Enterprise Content Manager sources of records

2.2 System architecture

In this section, we describe the system architecture of Enterprise Records and review the major components that make up the application.

2.2.1 Enterprise Records for IBM Content Foundation

The key to understanding Enterprise Records system architecture is to first understand the architecture of the underlying platform, IBM Content Foundation. This foundation includes full content lifecycle and document management capabilities to promote enterprise-wide content management (ECM) adoption and application development that uses existing and new types of content. Content Foundation includes the following components:

- ▶ IBM Content Platform Engine
- ▶ IBM FileNet Content Federation Services
- ▶ IBM Content Search Services
- ▶ IBM Content Management Interoperability Services for FileNet Content Manager
- ▶ IBM Content Navigator
- ▶ IBM FileNet Collaboration Services
- ▶ IBM Content Foundation documentation

Content Platform Engine is a FileNet P8 component that is designed to handle the heavy demands of a large enterprise. It can manage enterprise-wide workflow objects, custom objects, and documents by offering powerful and easy-to-use administration tools. Using these tools, an administrator can create and manage the classes, properties, storage, and metadata that form the foundation of an enterprise content management (ECM) system.

The key architectural aspects of the Content Platform Engine include:

- ▶ Object-oriented, extensible metadata model

Enables Content Platform Engine to provide complex and flexible data representation; and a rich event framework provides the means to trigger an action in response to activities performed against Content Platform Engine objects.
- ▶ Application programming interfaces (APIs)

Provide an extensible platform for development and cross-object store queries, and lets administrators configure systems programmatically. A Java-based API provides a rich set of Java classes that map to object store objects, such as Document, Folder, or Property Description; a Web Service API enables customers to write applications in a platform and language independent manner by exposing the object model in a small number of generic methods suitable for deployment in a web environment; a Microsoft .NET framework-based API, functionally equivalent to the Java-based API, provides for development of applications using the .NET framework.
- ▶ Java EE-compliant application server

Java Platform, Enterprise Edition (Java EE) offers reliability, scalability, and high availability features, and support for a wide range of operating system platforms, application servers, and database technologies.
- ▶ Scalable

Can be deployed to suit the demands of the enterprise. As the enterprise's needs change, you can reconfigure the system by replacing, adding, or

removing servers or applications without bringing the system down. You can add members to web server clusters and Content Platform Engine server clusters at any time.

► Unicode-based

Unicode is a universal character encoding standard that enables Content Platform Engine to be targeted across multiple platforms, languages, and regions without re-engineering, and it enables data to be transported through many different systems without corruption.

Figure 2-3 provides a high-level system overview of the Content Platform Engine architecture. It shows the relationship between the platform and where it resides within a typical *n*-tier distributed architecture:

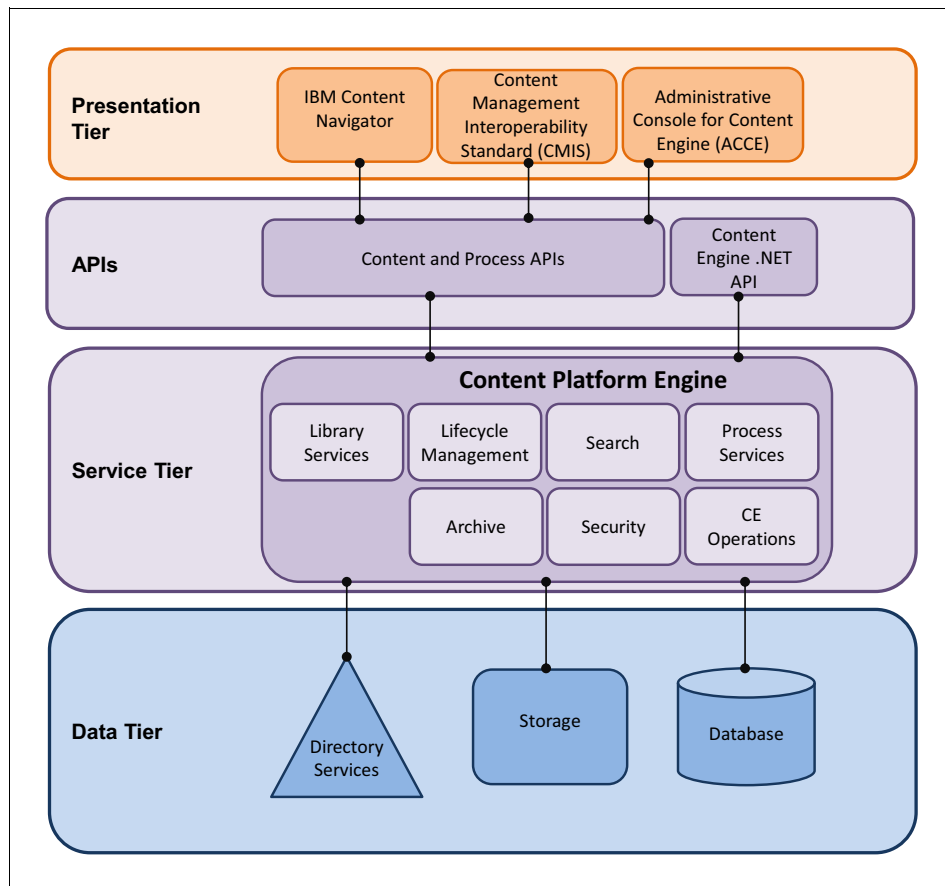


Figure 2-3 High-level architectural overview of IBM Content Foundation

Enterprise Records builds on top of IBM FileNet Content Platform Engine. It extends the services provided by the core Content Platform Engine and extends the Content Navigator interfaces. Enterprise Records provides records management functions, with a single repository that stores all electronic documents and records. With Enterprise Records, you can automate the management of electronic and physical records at the enterprise level.

Figure 2-4 shows several of the major Enterprise Records components within the IBM FileNet Content Foundation architecture and the relationships of those components to the underlying FileNet Content Platform Engine services.

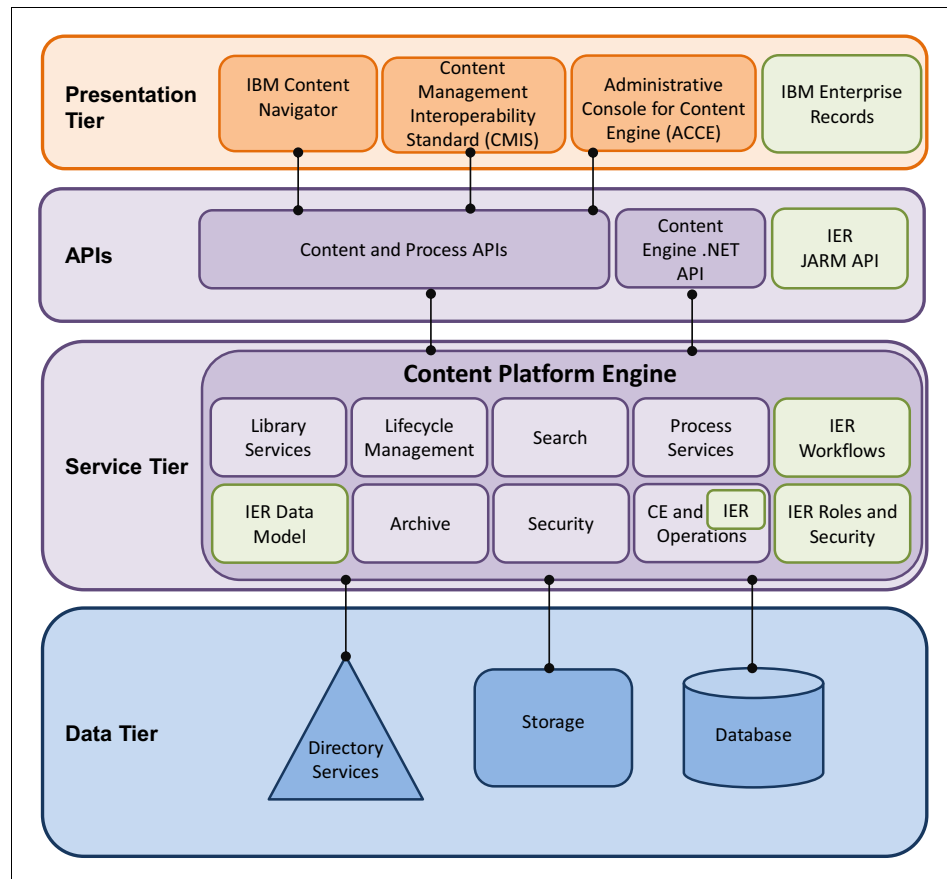


Figure 2-4 Architectural overview, IBM Content Foundation with IBM Enterprise Records

As Figure 2-4 on page 44 shows, the following are among the major Enterprise Records components:

- ▶ IBM FileNet Content Platform Engine:
 - Enterprise Records data model

This component provides the core definitions for Enterprise Records business objects, such as record classes, record folders, and disposition schedules, which is form the foundation of the records management system.
 - Enterprise Records roles and content engine security

The records management security capabilities are built upon the underlying Content Platform Engine security model coupled with default Enterprise Records security roles that determine functional user access. Enterprise Records uses Content Platform marking sets to implement certain features of its security model.
- ▶ Workflow features of the Content Platform Engine:

Process services provide special records management-related workflows for implementing a variety of disposition actions. Workflows use the full capability of the Process Service and can be completely customized.
- ▶ Within Content Navigator:
 - Enterprise Records application

Based on Content Navigator technology, the Enterprise Records interface has been redesigned, offering a role-based interface that enables customers to more easily install, manage, and use records management throughout the enterprise.

This component provides core administrative and management functions for the file plan and the records that it contains.
 - Content Navigator desktop

This component provides a comprehensive user interface and user access to documents and other business objects and access to any associated business processes. The Enterprise Records plug-in for Content Navigator gives users access to Enterprise Records features for manual record declaration.
 - IBM Java API for Records Management (JARM)

This API is a replacement for the original records management API and is intended for use in new custom Enterprise Records development solutions. It is currently used by the Content Navigator plug-in component, the Enterprise Records REST service API, and by several Enterprise Records tools.

- Component integrator (Enterprise Records Manager operations)

This component integrates records management functions into a workflow environment to work with business processes.

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

2.2.2 Relationship between content and records

The IBM FileNet family of products uses a common object model (or data model), managed by Content Platform Engine, that uses *object-oriented* design to store and manage content. Enterprise Records is built onto Content Platform Engine and, therefore, inherits the underlying object model and object-oriented design of Content Platform Engine. Information stored and managed in the system is 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 Content Foundation content repositories, which are also known as *object stores*. The object stores are managed by the Content Platform Engine.

For Enterprise Records configuration, there are two types of object stores:

- ▶ Record-enabled object store

The record-enabled 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 2-5 illustrates Enterprise Records integrated with the Content Foundation. It shows the relationship between records (which store the record-related metadata of the declared documents) in the FPOS and the associated declared documents in the ROS.

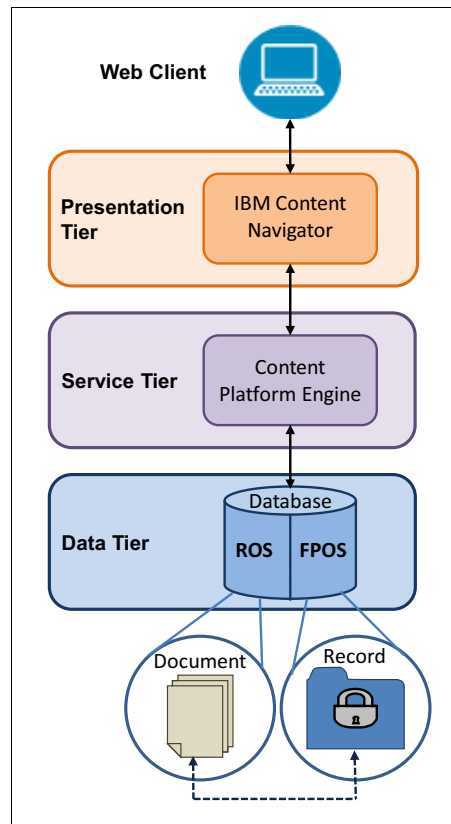


Figure 2-5 Overview of the relationship between a ROS and an FPOS

The diagram shows that an Enterprise Records record is a completely separate object from the associated declared document. The record must be contained in a file plan within the 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 within a Content Foundation content repository, the repository is simply another object store that is records-enabled and is managed by the Content Platform Engine. Other electronic documents can be stored outside of a Content Foundation content repository.

Physical records in Enterprise Records serve as markers or references to the physical objects stored outside of the system and contain information to identify and manage the physical objects in their home or transitory location.

This architecture enables Enterprise Records to easily unify and manage records across disparate, heterogeneous repositories, such as these:

- ▶ Native FileNet Content Foundation electronic documents
- ▶ Non-native electronic documents (external repositories)
- ▶ Physical records or artifacts

2.3 Data model, workflow, and security

In this section, we introduce data model options in Enterprise Records, the available workflows, and briefly describe Enterprise Records security roles.

2.3.1 IBM Enterprise Records data model

Content Platform Engine provides an object-oriented content repository; everything in the repository is represented as an object with associated metadata. Documents, folders, custom objects, and other business objects are all objects within the content repository. Enterprise Records entities are business objects defined in and managed by Content Platform Engine.

A key component in Enterprise Records is the data model. Using the object-oriented design of the Content Platform and extending its object model, the Enterprise Records Manager data model provides the core definitions that support the implementation of 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 FileNet Records Manager system.

Data model

Enterprise Records supports the following data model options:

- ▶ Base
- ▶ Department of Defense (DoD)
- ▶ DoD classified

The *Base data model* is the core Enterprise Records data model option that deploys the IBM recommended standard records management functions. This Base data model is the data model used by most IBM clients unless they require the specialized functions of another data model. For federation, Enterprise Records 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. Consider this option only if specific DoD-related configuration is required. (*DoD* refers to the Department of Defense standard 5015.2 for records management software. Each chapter in DoD addresses separate levels of the standard.)

The *DoD Classified data model* implements the functions specified in classified records chapter of the DoD standard. DoD Classified addresses requirements and functions for the management of classified records. Consider this data model only if you require Security Classification support.

Business objects in FPOS

The Enterprise Records data model encapsulates the underlying records management business objects and associated metadata. The business objects and their associated metadata are defined within the FPOS.

High-level records management business objects include these components:

- ▶ File plan
- ▶ Record containers:
 - Record categories
 - Record 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.

Figure 2-6 provides a high-level overview of some of the Enterprise Records data model and the potential relationships between the objects.

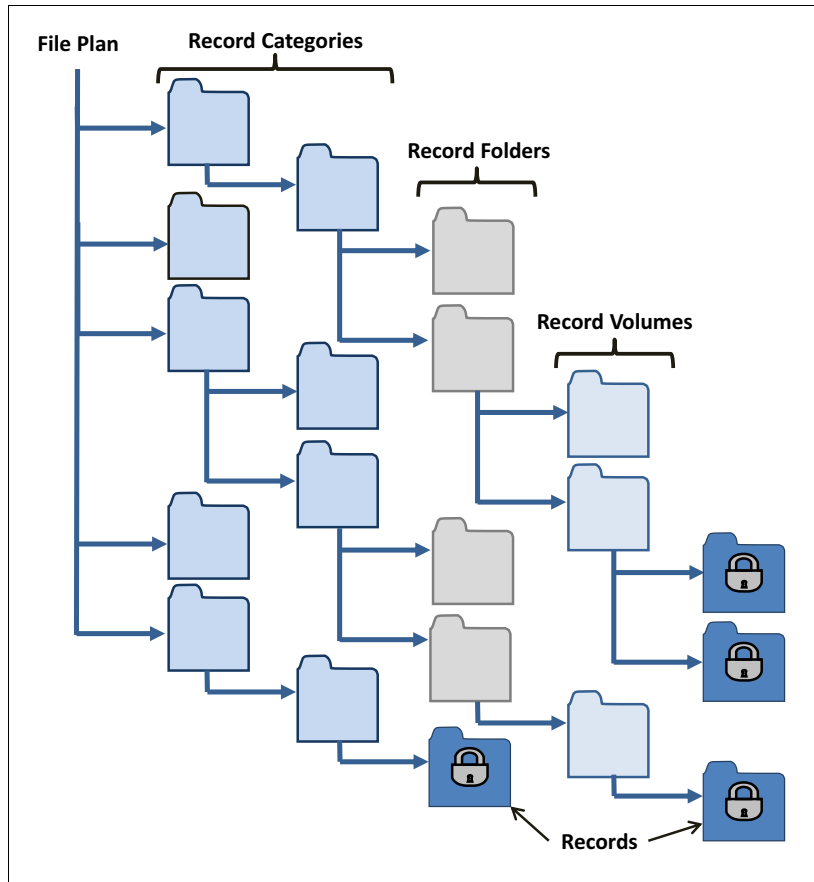


Figure 2-6 Sample IBM Enterprise Records data model

As shown in the diagram, at the top level of any data model there is a *file plan*, which can contain record categories. A *record category* can contain other record categories, record folders, or records. A *record folder* can contain *record volumes*, which contain records.

Disposition schedules and *disposition events* are not illustrated in Figure 2-6 but might be applied to the objects in the diagram. For instance, a basic disposal schedule might be applied to a record category.

Searches, also not shown in the diagram, can be defined and applied to most data model components, such as searches for record categories, record folders, and records.

Holds in Enterprise Records are applied to data model components, such as record categories, record folders, and records, to ensure that records are preserved in accordance with the hold requirements.

Security classification guides provide the details that govern classification when records administrators apply a classification guide to a file plan component.

2.3.2 IBM Enterprise Records workflows

One of the major benefits that Enterprise Records provides to organizations is that it is built on FileNet Content Platform Engine, which includes the strengths of the engine's Process Services. Like many areas of a business, records management is based on processes. Therefore, Enterprise Records is strongly process-centric.

To perform records management tasks, such as disposing of records, reviewing records before they are destroyed, destroying records, or working with physical records, Enterprise Records provides the following preconfigured workflows:

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

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

2.3.3 IBM Enterprise Records security and roles

A key feature 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 to the document that were specified before it was declared a record. You also can use *inherited proxy* so that documents retain the security that they had before they were declared, except that the document is locked down and cannot be deleted except by Enterprise Records.

Enterprise Records 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 the following safeguards:

- ▶ 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, which are recorded in the audit log. Records users cannot delete records accidentally.
- ▶ Users can perform only those operations on records for which they have access rights granted.

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

Enterprise Records supports several 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. These standard roles are defined by the Base data model:

Records Administrator	Users who are Enterprise Records system administrators.
Records Manager	Users who are responsible for applying disposal schedules, holds, managing the file plan and file plan components, physical records, and disposition processes.
Records Privileged User	Users who have delegated access to a segment of file plan components for a business unit or department. In some organizations, this is the departmental Information Coordinator.
Records User	General users who might require access to Enterprise Records user functions. Within most organizations, it is typical for general users to never interact with an Enterprise Records desktop and to work with Content Navigator desktops instead.

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

The predefined roles vary with other data models. For example, the DoD Classified data model implements an additional role called the Classification Guide Administrator.

For a more information about these roles and security, see Chapter 4, “Security” on page 93.

2.4 Logging

Within Enterprise Records, logging is typically implemented to collect and record information about application failures in test or production environments. When you enable logging, log statements inserted in to shipped Enterprise Records application code cause log entries to be written to an output location.

As the application makes requests to the server, the logging mechanism captures information about the request. The mechanism writes the information to some output medium, such as a file or a console. The information can then be used for analyzing and debugging problems. The information that you capture and its level of detail are controlled by modifying the configuration file in Enterprise Records. This file provides a way to log behavior, priority levels, and output formats with no need to change or recompile the code in the application.

2.5 User and administrative applications

Enterprise Records has been redesigned and supports role-based user interfaces, based on Content Navigator desktops, to enable customers to more easily deploy, manage, and use records management throughout the enterprise. The administration client gives you the following set of features and enhancements over the traditional Enterprise Records client:

- ▶ Ability to dispose of excess data by improved integration with the IBM Atlas Policy Suite:
 - Protection of externally managed objects by using “Global Retention Policy and Schedule Management” syndicated schedules
 - Improved granularity of controls for external management by using Atlas Policy Suite containers and schedule mapping
- ▶ Improved efficiency and usability in the entire user interface:
 - Improved browse, navigation, and management of file plans and file plan containers

- Improved records administration in the areas of records creation, retrieval, update, and deletion
- Improved user interface for basic schedule disposition sweep and report execution
- Improved Content Navigator technology with the addition of Favorites and External Data Service support

2.5.1 IBM Content Navigator

Content Navigator is a web client that provides users with a console for working with content from multiple content servers. It also enables users to create custom views of the content by creating team spaces, which provide a focused view of the relevant documents, folders, and searches that a team must complete their tasks. Content Navigator provides a single user interface for access to content, collaboration, workflows, and records.

Content Navigator desktops are the primary mechanism by which day-to-day records users interact with Enterprise Records (if no custom applications are created). The degree to which users are aware of Enterprise Records is, in part, dependent upon whether you want to expose the application to users. Preferred practices typically dictate minimizing or completely masking the users' knowledge of the underlying Enterprise Records system. Or you can set up Enterprise Records to be completely visible to users through automated declaration technology.

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

Figure 2-7 on page 55 provides an overview of an Content Navigator desktop which is records-enabled. As the desktop is records-enabled, it contains additional context menu options that enable users to declare documents as records and to view record properties. The availability of a user to use a feature is dependent upon the user having appropriate permissions, and on the context of feature. In the figure, the document has not yet been declared as a record, so the user has permission to declare the document as a record, but the user is unable to view the document's record properties.

Note the folder structure in Figure 2-7 on page 55 is there for illustrative purposes only. The choice as to whether to use folders or not in the record object store is dependent upon your business requirements, and might differ between departments or lines of business.

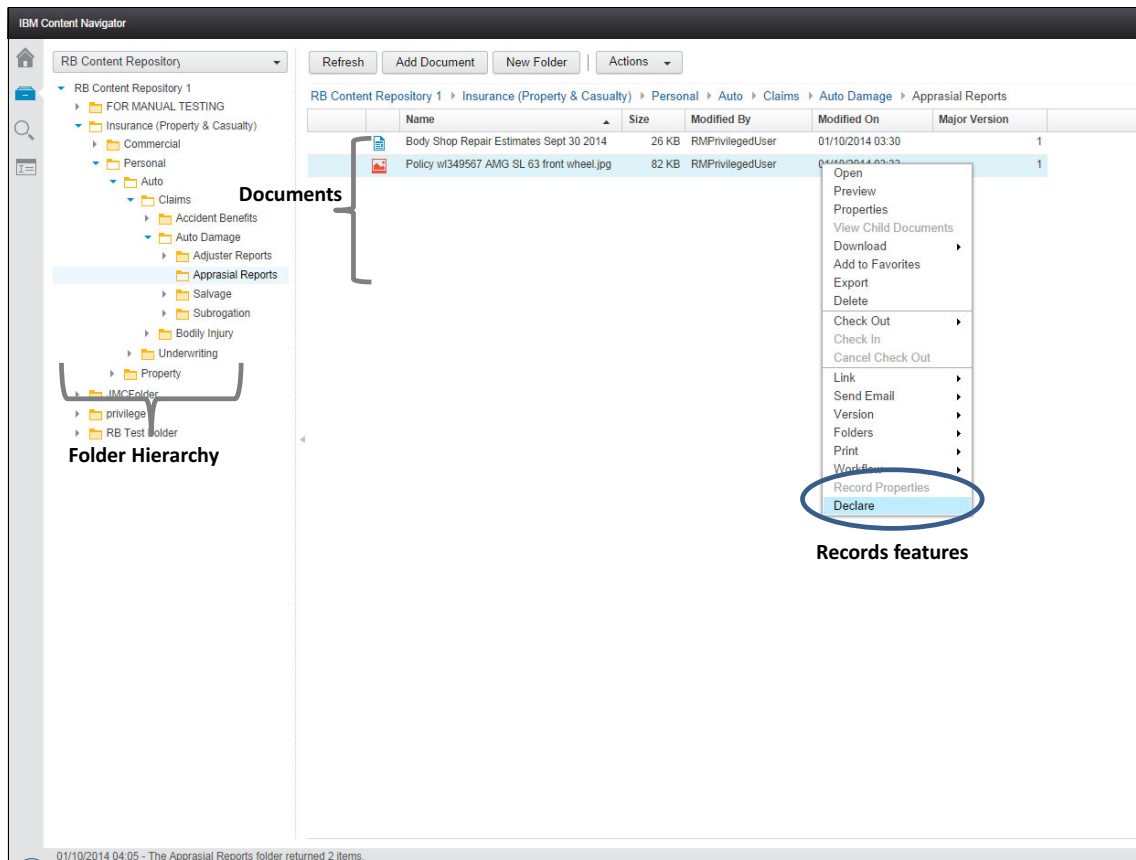


Figure 2-7 Sample IBM Content Navigator records-enabled desktop

Within Enterprise Records, it is possible to control how users interact with Enterprise Records. For some organizations, it might be desirable or necessary in certain situations to give users control over the record declaration process, enabling the user to choose when to declare a document as a record. Enterprise Records enables you to configure various levels of control:

- ▶ 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 Enterprise Records controls where to declare them as records
- ▶ No control, where the record declaration process is completely automated and is apparent to users

Content Navigator also includes a powerful API toolkit that you can use to extend the web client and build custom applications.

2.5.2 IBM Enterprise Records interface

The Enterprise Records administrative client application is the key user interface. It 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, Records Administrators, and privileged users, such as departmental Information Coordinators, although it can also be used by other users for search and retrieval and to perform other manual operations.

Figure 2-8 shows the Enterprise Records Content Navigator user interface.

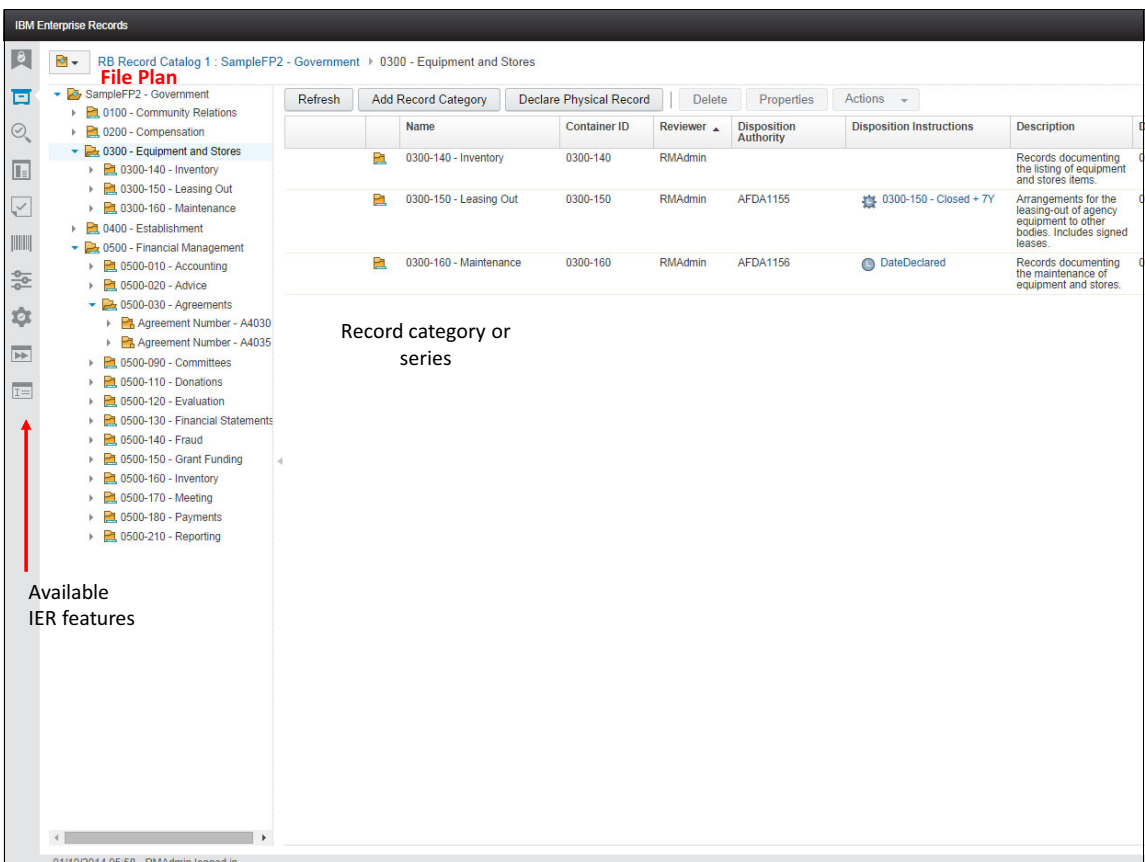


Figure 2-8 Sample IBM Enterprise Records desktop for Records Administrators or Records Managers

On the left side of the interface are the available administration and management options, which include the following features:

- ▶ Favorites. This is where users might save their most widely used activities, such as browsing a particular category or record folder in the file plan or their favorite searches.
- ▶ Browse File Plan. This provides the records management user with access the available file plans, which might be selected from a drop-down menu and then explored.
- ▶ Open Search View. The search view is where the records management user might create, manage or run searches from. This view maintains a list of recent searches. This will be covered further in the next section.
- ▶ Open Reports View. This view provides the records management user with access to the ready-for-use and custom reports available for the system. The reports accessed in this interface are IBM Cognos BI reports, and the following reports are included:
 - Actions performed by a user
 - Containers without an associated disposal schedule
 - Electronic record contents viewed by a user
 - Records placed on hold
 - File plan structure
 - Items associated with a disposition schedule
- ▶ Open Task View. This view provides access to the Enterprise Records Task Manager. The Task Manager service is a REST web application that can be used to run background or asynchronous tasks for any repositories. The Task Manager enables you to create and run automated and scheduled tasks. You can also use it to run large tasks in the background while you perform other tasks.

The Task Manager is a Java Platform, Enterprise Edition 6 compliant web application.

The Enterprise Records Task Manager is used to schedule and run tasks such as these:

- Reports
 - Basic schedule tasks
 - Advanced schedule tasks
 - Hold Sweep tasks
- ▶ Open Physical Items View. The physical items view provides records management users access to manage and work with physical items under records management.

- ▶ **Open Configuration View.** This view is primarily for the Records Manager and Records Administrator to configure or define elements of the system used to manage records, which include:
 - Holds
 - Locations
 - Naming patterns
 - File plans
 - Report definitions
 - Advanced disposal schedules
 - Actions
 - Event triggers
- ▶ **Open Administration View.** This view provides access to system level configuration of Enterprise Records, such as these:
 - Columns displayed in the desktop
 - Report server settings
 - Settings for performance tuning

This feature is typically configured to be available only in the Enterprise Records Administrator's desktop.
- ▶ **Open Work View.** This view provides Records Managers or Records Manager privileged users, such as the departmental Information Coordinator, access to outstanding workflow tasks that require action. It gives these users access to items allocated to them as the individual who is responsible for a record collection, or set of record collections, and access to general workflow queues that the user has authorization to process.
- ▶ **Open Entry Templates View.** This view provides access to create record entry templates for manual declaration of records. Access to this feature is typically provided to Records Manager privileged users, such as the departmental Information Coordinator.

The preferred practice is to create different Content Navigator desktops for the various records management users. This provides access to only those features that are required to perform the tasks of the role.

Ad hoc search

The search feature in Enterprise Records gives authorized users access to define and run new (ad hoc) searches, rerun recent searches, and work with saved (stored) searches.

Figure 2-9 shows the ad hoc search interface that enables users to define what the context of the search is, the types of entity they are searching for, specific metadata and values to look for, whether the values should be joined by Boolean “and” or “or” operators, and the layout of the search results, for example which value to use to sort the results.

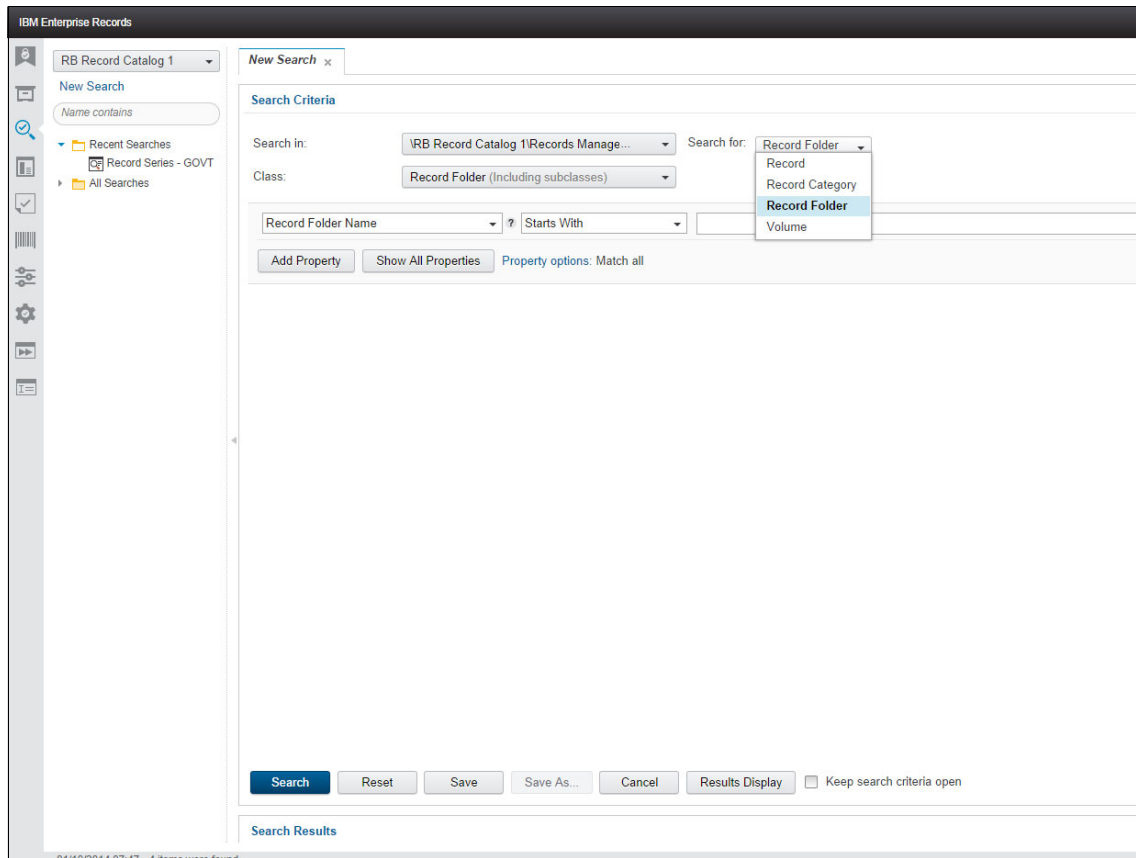


Figure 2-9 Sample ad hoc Search window

When creating an ad hoc search using this interface, you can save the search to use again.

Stored searches

All search criteria for a stored search has been defined by the person who created the search. All you do is click the name. This action runs the search and presents the results.

Expanding the section labeled **All Searches** (see Figure 2-11) displays a list of all stored searches that are available to the user.

When creating a stored search, it is possible to leave the search criteria open so the user can modify the value of the search criteria. In this case, the saved search functions as a search template.

2.5.3 Disposition process

Enterprise Records now supports two types of disposition sweeps:

- ▶ Basic schedule disposition sweep
- ▶ Advanced schedule disposition sweep

Both of these disposition sweeps are accessed from the Enterprise Records administrative client by using the Task Manager. These sweeps primarily update information about record entities. They are also responsible for automatically initiating several of the optional workflows, such as the Screening workflow and the Cutoff workflow.

Preferred practice for scheduling these sweeps is for the Records Manager to work with the Records Administrator to ensure the sweeps are run at times that will not affect other system administration tasks or system performance for users.

For more information about disposition, see Chapter 6, “Records disposition and basic schedules” on page 151, and Chapter 7, “Advanced disposition” on page 175.

2.5.4 Hold process

Hold Sweep in Enterprise Records is accessed from the Task Manager in the administrative client. The Hold Sweep finds records that meet the conditions specified in one or more conditional holds and automatically places these records on hold. A *conditional hold* enables you to specify metadata-based search criteria that you can use to evaluate whether a record needs to be placed on hold.

Typically, Hold Sweep 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 added and declared as records, several of the new documents might also require being 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. This frees the records management staff from having to periodically search for these documents and manually place them on hold.

For more information about holds, see Chapter 8, “Holds and preservation” on page 213.

2.5.5 IBM Administration Console for Content Engine

The Administration Console for Content Platform Engine (ACCE) is a web-based tool for configuring and administering content, workflow, and analysis features in Content Platform Engine. The administration console replaces Enterprise Manager as the primary administration tool for Content Platform Engine.

2.5.6 File Plan Import Export Tool

The File Plan Import Export Tool is a stand-alone application that enables 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 file plan object stores (FPOSeS), if required.

2.6 APIs and the Component Integrator

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

2.6.1 IBM Enterprise Records and Bulk Declaration Services

There are two API sets available for custom application development related to Records Manager functions:

- ▶ IBM Java API for Records Management (JARM)
- ▶ Bulk Declaration Service (BDS)

Java API for Records Management

Enterprise Records provides an application programming interface (API), Java API for Records Management (JARM) that exposes all of Enterprise Records Manager for custom application development. JARM is a replacement for the original Records Management API and is intended for use in developing new custom Enterprise Records solutions.

Bulk Declaration Service API

The BDS API is available for the high performance, large volume ingestion of records into Enterprise Records. 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 Content Foundation repository.

BDS provides the following functions:

- ▶ Bulk declaration of new physical records
- ▶ Bulk declaration of new electronic records for existing documents in the Content Foundation repository
- ▶ Bulk creation of new documents in the Content Foundation repository and optionally declaring them as records

2.6.2 IBM Enterprise Records Component Integrator

Enterprise Records uses a workflow-centric approach to compliance and records management. The Enterprise Records Component Integrator tool provides a mechanism for record-enabling business workflow. The Component Integrator makes it possible to import custom Java components as code modules to make them available in a workflow. In addition, a Java Platform, Enterprise Edition Java Message Service (JMS) queue can be configured as a component queue where the work object, in XML format, is sent to the JMS queue when the work item is routed to the component queue in the workflow.

In the workflow definition, a component step connects to a component queue that is configured for one or more operations in the external component.

The Component Integrator consists of two parts:

- ▶ The Component Manager. This service runs on the Content Platform Engine and connects a work item with its appropriate Java or Java Message Service (JMS) adapter.
- ▶ Configuration. Configure the component queues by using Process Configuration Console.

The Enterprise Records Component Integrator integrates directly with the workflow modules to provide records management capabilities within workflows, including these actions:

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

You can extend the integration of Enterprise Records and the workflow module by building custom Java components, using JARM to expose more specialized and tailored records management capabilities directly to workflows.

2.7 Reporting

Enterprise Records includes ready-for-use support for IBM Cognos BI reporting and for SAP Crystal Reports. Reporting is supported through a reporting framework that provides a set of Report Engine application programming interfaces (APIs). Use the Report Engine API to run Content Platform Engine queries and place the results in a database table. A reporting application such as Cognos can extract the information from the database and generate a report that is based on a template. A set of predefined templates are provided for Enterprise Records reports.

Configuration of the reporting server, schedule of reports, and creation of new reports are all accessible from within the administrative client.

For more information, see Chapter 10, “Reporting” on page 243.

2.8 References

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

- ▶ IBM Enterprise Records web page
<http://www.ibm.com/software/products/en/entereco>
- ▶ IBM Enterprise Records Support Portal
<http://ibm.co/10UIPMk>
- ▶ IBM Enterprise Records V5.1.2 documentation
<http://ibm.co/1ce9u1U>

- ▶ IBM Enterprise Records Publication Library
<http://www.ibm.com/support/docview.wss?uid=swg27042282>
- ▶ Forum on IBM developerWorks®
<http://ibm.co/1F211S4>



Retention 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 Enterprise Records concepts.

This chapter is organized into sections that cover the following topics:

- ▶ Retention schedule
- ▶ Retention schedule planning and creation
- ▶ File plan
- ▶ File plan planning and creation
- ▶ File plan in IBM Enterprise Records
- ▶ File plan case study

Disclaimer: We provide the information in this chapter *as is* without warranty of any kind, expressed or implied. IBM is *not* responsible for any damages that might arise out of the use of, or otherwise related to, this information. Nothing included in this book is intended to, nor shall have the effect of, creating any warranties or representations from IBM (or its suppliers or licensors), or altering the terms and conditions of applicable agreements governing the use of IBM hardware, software, or services.

3.1 Retention schedule

A *retention schedule* is a timetable that specifies the length of time that a record must be retained before destruction. A retention schedule describes a company's regulated records, their ownership, regulatory citations, and the retention period based on legal, regulatory, and business needs. The company needs to review the retention schedule from time to time and to dispose of records in accordance with the retention schedule.

Most companies have some form of retention schedule or *retention rules* for their records. If your company has a retention schedule, it should be reviewed and revised according to requirements that we describe here. If you do not have a schedule yet, you must create one.

A retention schedule can be guided by the following requirements:

- ▶ *Compliance* and *regulatory* requirements. Industry and government regulations often impose different retention requirements for records.

Terminology: *Compliance* is the act of adhering to and demonstrating adherence to internal or external regulations. A *regulation* is a compromise between prohibition and no control at all.

- ▶ *Fiscal* requirements related to recordkeeping.
- ▶ *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 from different 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 as to 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 such laws.

IBM does *not* provide legal, accounting, or audit advice or represent or warrant that its services or products will ensure that a client 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 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 guidelines to create a retention schedule, as shown in Figure 3-1 on page 68:

1. Develop a records management policy.
2. Specify records management procedures.
3. Conduct a records inventory.
4. Understand record *series*.
5. Record and update regulatory requirements.
6. Compile retention schedules.
7. Implement file plans.

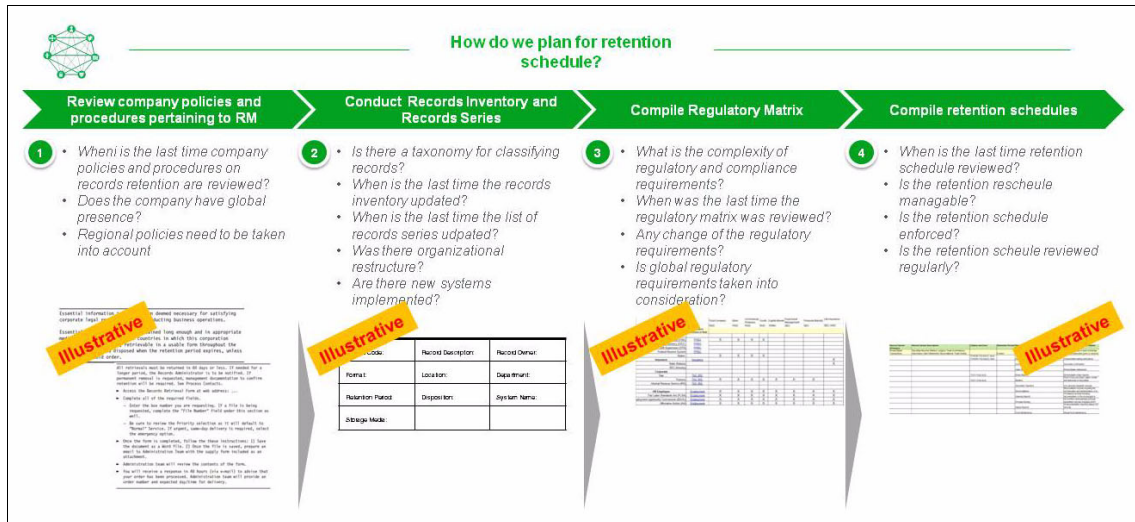


Figure 3-1 Planning a retention schedule

3.2.1 Develop a 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, a preferable practice is to develop one. A policy is a living document and should be reviewed and revised as business need evolves. In addition, everyone in the company must 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.

As shown in Figure 3-2 on page 69, company policy is the cornerstone of a records management program. It has a direct impact on the processes and procedures for the program.

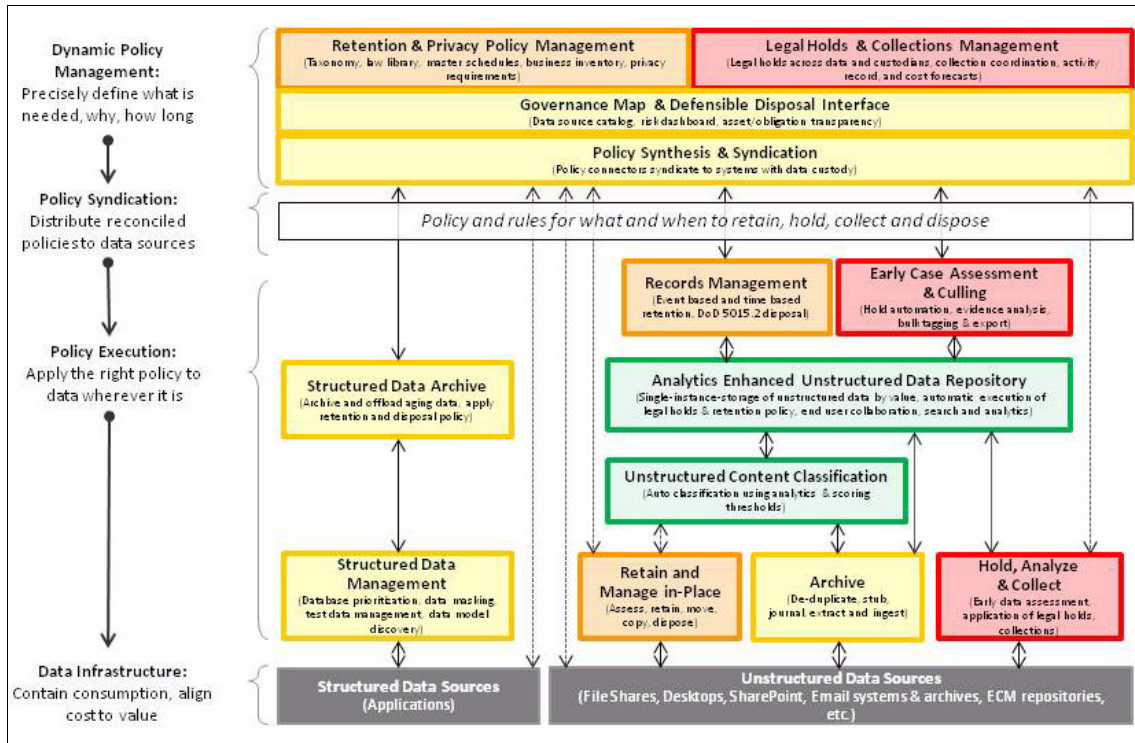


Figure 3-2 Policy is the cornerstone of ILG governance

Example 3-1 shows 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.

Often, a company's records retention policy is affected by global retention requirements. In both the United States and Canada, electronic records are usually acceptable when records are legally required to be kept.

For example, accounting records are required to be kept for six years in accordance with Securities and Exchange (SEC) Act Rules 17a-3 (17cfr240.17a-3(a)(2)) in US.

The record management requirements of Asian Pacific countries seem to parallel many of the general principles used in other regions. For most Asian countries and countries throughout the world, if a company is registered in that country, it is normally obligated to maintain accounting records for 10 years.

As explained previously, most US-based multinational companies retain many accounting and general business records for retention periods ranging from five to seven years. However, as indicated above, many global retention requirements applicable to these records specify 10-year retention periods.

One way to implement the records retention policy is to retain the accounting record for the retention period for most accounting records unless law and practice necessitate a longer period.

3.2.2 Specify 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 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 company's industry, the nature of the business, and the operation of each area within the company.

Example 3-2 shows part of a records retrieval procedure.

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 as 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:
 - a. Save the document as a Word file.
 - b. After the file is saved, prepare an email to 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 (through email) to advise that your order has been processed. Administration team will provide an order number and expected day/time for delivery.
-

3.2.3 Record and update regulatory requirements

Many countries around the world have laws about recordkeeping. Most are applicable to physical and electronic records, some specify the active and inactive retention period, and some have special compliance requirements for storage.

Regulatory requirements vary from country to country, industry to industry, legal entities, states, and document types. See 1.5.1, "Addressing regulatory requirements" on page 10, for an overview of regulatory requirements that are pertinent to a financial institution in the US.

3.2.4 Conduct a records inventory

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

Table 3-1 on page 72 shows an example of a records inventory template.

Note: The list that we show here is not meant to be exhaustive.

Table 3-1 Sample record inventory template

Record Code:	Record Description:	Record Owner:
Format:	Location:	Department:
Retention Period:	Disposition:	System Name:
Storage Media:		

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

Review the existing records filing process

Many companies already have some form of records filing processes. Review and study the existing record 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 between groups of records
- ▶ Ensure that all records from the company are presented and collected
- ▶ Identify who created the records and how they were created

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

3.2.5 Define records series

Use a single consistent taxonomy, globally, across records, business, and information technology.

Terminology: A *record series* is a group of related records grouped as a unit and evaluated as a unit for retention and disposition purposes.

Figure 3-3 on page 73 shows an example of a group of accounting records, which can include payroll records, accounts payable, accounts receivable, general ledger, tax records, and so on. All of these are similar in nature and have similar retention requirements.

Entity	Record Series (Primary)	Record Series Description	Citation and time	Retention Period	Record Sub-Series/ Secondary Classification
Bank	Accounting	Registers: Fixed Assets, Employee Payroll Accounting, Payroll	12CFR228.38(g)	7 years	Accounts Payables
Bank	Accounting	Registers: Fixed Assets, Employee Payroll Accounting, Payroll	12CFR228.38(g)	7 years	Accounts Receivables
Bank	Accounting	Registers: Fixed Assets, Employee Payroll Accounting, Payroll	12CFR228.38(g)	7 years	Balances
Bank	Accounting	Registers: Fixed Assets, Employee Payroll Accounting, Payroll	12CFR228.38(g)	7 years	Bank cash management
Bank	Accounting	Registers: Fixed Assets, Employee Payroll Accounting, Payroll	12CFR228.38(g)	7 years	Financial Statements
Bank	Accounting	Registers: Fixed Assets, Employee Payroll Accounting, Payroll	12CFR228.38(g)	7 years	General Ledger
Bank	Accounting	Registers: Fixed Assets, Employee Payroll Accounting, Payroll	12CFR228.38(g)	7 years	Subsidiary ledgers
Bank	Accounting	Registers: Fixed Assets, Employee Payroll Accounting, Payroll	12CFR228.38(g)	7 years	Journals
Bank	Accounting	Registers: Fixed Assets, Employee Payroll Accounting, Payroll	12CFR228.38(g)	7 years	Tax Reporting
Bank	Accounting	Registers: Fixed Assets, Employee Payroll Accounting, Payroll	12CFR228.38(g)	7 years	Treasury Secured Financing
Bank	Accounting	Registers: Fixed Assets, Employee Payroll Accounting, Payroll	12CFR228.38(g)	7 years	Treasury Credit Support Records
Bank	Accounting	Registers: Fixed Assets, Employee Payroll Accounting, Payroll	12CFR228.38(g)	7 years	Investment Management
Bank	Accounting	Registers: Fixed Assets, Employee Payroll Accounting, Payroll	12CFR228.38(g)	7 years	Employee Stock/Purchase Plans
Bank	Accounting	Registers: Fixed Assets, Employee Payroll Accounting, Payroll	12CFR228.38(g)	7 years	Collections
Bank	Accounting	Registers: Fixed Assets, Employee Payroll Accounting, Payroll	12CFR228.38(g)	7 years	Billing
Bank	Accounting	Registers: Fixed Assets, Employee Payroll Accounting, Payroll	12CFR228.38(g)	7 years	Deposit and check

Figure 3-3 Extract of a sample records series

By doing this, you can catalog the classes and types of information within the organization and specify where and how it should be managed. You can easily enable country and local schedule variations as needed and compare retention practices for the same information across the business and systems.

In the next section, we show how to overlay a regulation matrix over records series.

3.2.6 Create a regulatory matrix

Catalog the laws and regulations that stipulate both retention and privacy requirements, along with policies for security.

Apply laws to information to enable precise retention and privacy actions. Associate specific laws and regulations with specific classes and jurisdictions to retain less data more defensibly.

Terminology: A *regulatory matrix* provides a comprehensive list of federal laws and regulations that govern the various entities of the company.

Figure 3-4 on page 74 is an extract of a sample of regulatory matrix for a typical financial institution.

Record Retention		Trust Company	Bank	Commercial Finance	Credit	Capital Market	Fund Asset Management	Financial Markets	Life Insurance
Regulatory Matrix		FDIC	FDIC	FDIC	FDIC	IFSR	SEC	SEC	SEC, NAIC
Business Specific	Regulatory Reference or Rule								
Banking									
Federal Deposit insurance Corp (FDIC)	FFIEC	X	X	X	X				
Office of the Comptroller of the Currency (OCC)	FFIEC								
Office of Thrift Supervision (OTS)	FFIEC								
Federal Reserve System	FFIEC								
States		X	X	X	X				
Insurance	Insurance								X
State Statutes									X
SEC (Annuity)									
Corporate									
Tax	TAX_IRS								
Treasury	TAX_IRS	X	X	X	X	X	X	X	
Internal Revenue Service (IRS)	TAX_IRS								
HR Employee	Employment	X	X	X	X	X	X	X	X
Fair Labor Standards Act (FLSA)	Employment	X	X	X	X	X	X	X	X
Employment opportunity Commission (EEOC)	Employment	X	X	X	X	X	X	X	X
Affirmative Action (AA)	Employment	X	X	X	X	X	X	X	X

Figure 3-4 Regulatory matrix

Global presence

Multinational companies have business operations, offices, and records throughout the world. As companies become global, Records Managers must know which country's laws and regulations will affect their record management policies.

In our example, the Securities Exchange Commission (SEC) in the US requires companies to retain accounting records for six years. However, some other parts of the world, such as China and Japan, require accounting records and tax documents to be retained for 10 years.

Data residency concerns

As more companies take advantage of cloud storage, it's important to also look into data residency and sovereignty requirements. Some countries' regulations require companies that deal with personal information to restrict the transfer of personal information outside of their borders. This might require further consideration when implementing a records management system and the physical location of the data stores.

In the next section, we show how to put everything together and create a retention schedule.

3.2.7 Creating the retention schedule

After gathering a complete records inventory, categorize records into groups of related records that can be filed as a unit for retention purpose. Retention period for each group should be based on legal mandates, regulatory requirements, business requirements, and good business practices.

For example, SEC 17a-4 requires that some records retained by brokers and dealers must be preserved for at least six years, the first two years in an easily accessible place. Other records must be retained for at least three years, the first two years in an easily accessible place. Based on regulation 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 company's business operation perspective) *and* they all have the *same* retention period based on requirements.

In addition to determining the proper retention period for records, you must determine the disposal method for records when their retention period expires. This is important because proper disposition of records protects companies from future liability and controls information storage size. Records disposition options include (but are not limited to) destroying the records and archiving records to another records holding authority. You can also set up the system to review records when their retention periods expire and then decide what to do.

Note: By design, a records management system does not destroy records automatically when records reach the end of their retention periods, even when the disposition is set to *destroy*. The system always requires human verification before anything is done to the expired records.

Provide precise, actionable instructions to information holders about the applicable privacy and security obligations. Enforce them on unstructured and structured data automatically.

Also consider the following related requirements:

- ▶ Retention period
- ▶ Disposal method requirements
- ▶ Storage media requirements
- ▶ Security requirements
- ▶ Use limits
- ▶ Transport or transfer requirements
- ▶ Disclosure of failure requirements

Important: *Retention period* and *records disposition method* are two of the key elements in a retention schedule. Understanding your company's records policy and records procedures and understanding and *correct interpretation* of laws and regulations are crucial in determining the length of time that a particular group of records must be kept and what to do with them afterward. Legal counsel and special records professionals should be involved in developing the actual retention schedule for a company.

For more information about records disposition, see Chapter 6, "Records disposition and basic schedules" on page 151.

Table 3-2 is an excerpt from the retention schedule that we develop later, in Chapter 14, "File plan case study" on page 295.

Table 3-2 Excerpt from a retention schedule

Category	Description	Citation	Total retention period	Disposition
Corporate governance	Books and records that substantiate the existence, operation and obligations of each legal entity, such as: Articles of Incorporation; Board of Directors, Shareholders	NASD3510 life NASD3010(b)(3) life 17cfr270.31a-2(a)(1) permanent 17cfr270.38a-1(d)(1) life +5 years 17cfr275.204-2(e)2 5 years 17cfr240.17a-4(d) permanent	Permanent	Not applicable

Category	Description	Citation	Total retention period	Disposition
Broker-dealer transactions	Securities records; blotters; ledgers; trade confirmations; instructions; client statements, reconciliations; trade sheets, stock bond records	17cfr240.17a-3(a)(1), 6 years 17cfr240.17a-3(a)(2), 6 years 17cfr240.17a-3(a)(3), 6 years 17cfr240.17a-3(a)(4), 3 years 17cfr240.17a-3(a)(4)(i), 3 years 17cfr240.17a-3(a)(4)(ii), 3 years 17cfr240.17a-3(a)(4)(iii), 3 years 17cfr240.17a-3(a)(4)(iv), 3 years 17cfr240.17a-3(a)(4)(v), 3 years 17cfr240.17a-3(a)(4)(vi), 3 years 17cfr240.17a-3(a)(4)(vii), 3 years 17cfr240.17a-3(a)(5), 6 years 17cfr240.17a-3(a)(6)(i), 3 years 17cfr240.17a-3(a)(6)(ii), 3 years 17cfr240.17a-3(a)(7), 3 years 17cfr240.17a-3(a)(8), 3 years 17cfr240.17a-3(a)(9), 3 years 17cfr240.17a-3(a)(9)(iii), 3 years 17cfr240.17a-3(a)(10), 3 years 17cfr1.31(a) 5 years 17cfr1.31(d) 5 years	6 years	Destroy
Client communications	Correspondence, communications (email, voice recordings and postal mail), complaints from clients, both institutional and retail	17cfr275.204-2(e)(3) 5 years	5 years	Destroy
Employee data	Monitoring and reporting on all employee-related activity and activity of associated persons, such as employee files, fingerprinting, compensation or salary, benefits, registration, licensing	17cfr240.17a-3(a)(12)(i), employment +3 years, 17cfr240.17a-3(a)(19)(i), 3 years	Termination date + 3 years	Destroy

Category	Description	Citation	Total retention period	Disposition
Accounting	Financial statements, journal entries, general ledgers, P&L statements, documentation of fixed assets, employee payroll accounting records, payroll registers, tax returns, and working papers	17cfr270.31a-(b)(1) 2 years 17cfr270.31a-(b)(2) 2 years 17cfr275.204-2(e)(1) 5 years 17cfr275.204-2(e)(3) 5 years 17cfr1.31(a) 5 years 17cfr240.17a-3(a)(2) 6 years 17cfr240.17a-3(a)(3) 6 years 17cfr240.17a-3(a)(4) 3 years 17cfr240.17a-3(a)(11) NAIC 910-1 section 4, current +3 years	6 years	Destroy

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

Record series	A <i>group of related records</i> 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 <i>name</i> by which the group of records are known by the users.
Description	Defines the <i>scope</i> of the records included in the category.
Office of record	Refers to the <i>owner</i> , 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 if there is a disaster. Vital records are usually stored offsite and replicated for <i>disaster recovery</i> purposes.
Active retention period	The retention period for records that are required for <i>current</i> use.
Inactive retention period	The period of time during which inactive records must be maintained by the company.

Terminology: *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* indicates 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 active retention period and inactive retention period.
Citation	The statutory authority or law that governs the retention of the records. For example, SEC 17cfr275.204-2(e)(3) requires that client communications to be retained for five years.
Disposition	The final <i>action</i> for a records series. Examples of valid actions are <i>destroy</i> (physically destroying the records) and <i>accession</i> (or archiving, transferring records to other records holding authorities).
Medium	The object or device where the record resides. Examples of media are paper record, microfilm, computer disk, and CD-ROM.
Records series code	Can be used to refer to a citation schedule or a disposition schedule.

3.3 File plan

A *file plan* is an IBM Enterprise Records instance to implement the enterprise's retention schedule. It is a structured, *subject-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. It is different 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 managements policies.

Terminology: 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 who is searching for a piece of music knows that it is a concerto, the user can narrow the search to this category. If the user does not know that the piece is classical, the user must broaden the search.

In a typical company environment, there are different documents that can be associated with different buckets (or subject folders). Each folder is then tied to the retention rule, which indicates how long documents within the folders are to be kept until destruction.

You can apply retention rules at either the folders or records level, as shown in Figure 3-5 on page 81. A single record can include multiple documents or multiple pieces of information.

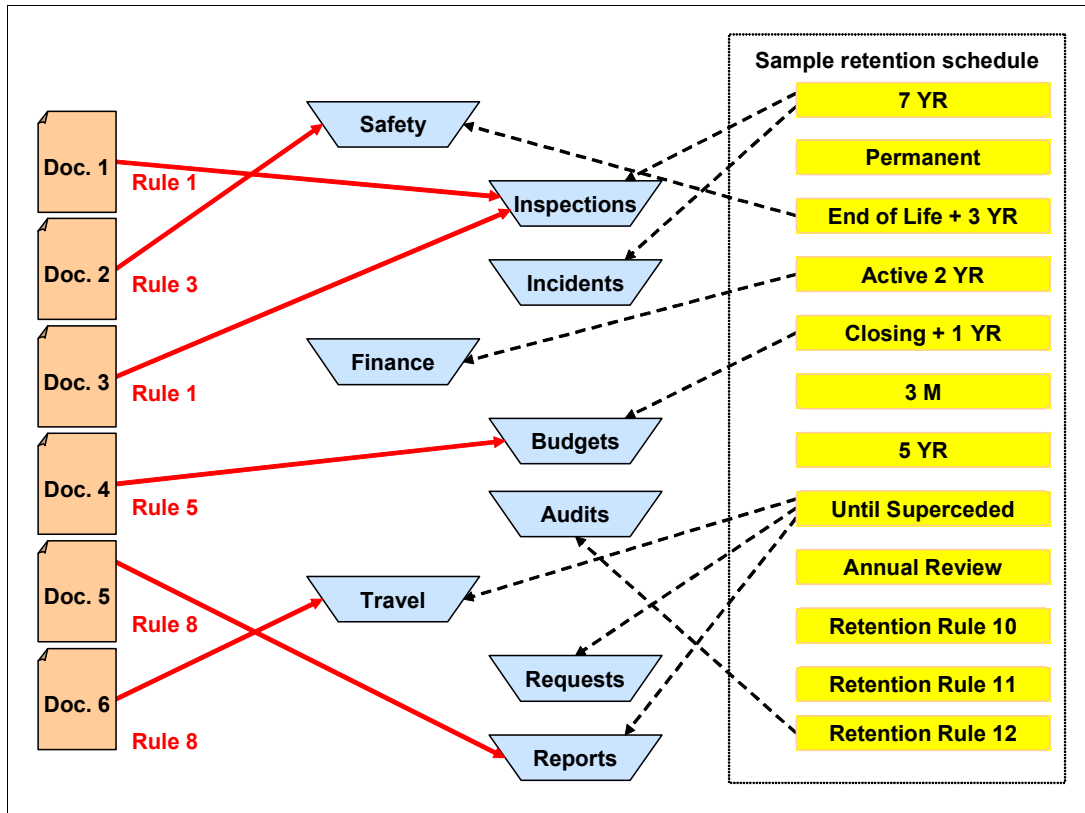


Figure 3-5 Classify a document according to a bucket (subject folder) that is tied to a retention rule

Terminology: *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 US Department of Defense (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.

3.4 File plan planning and creation

As mentioned in 3.2.4, “Conduct a records inventory” on page 71, many companies have some form of records filing processes. Before you design your file plan, review the records filing processes and records inventory that you generated when following that section. Design a file plan that models the records and their hierarchical relationships within the company.

Figure 3-6 shows an excerpt from an example of a structure for a fictitious global financial institution. The company has representations in multiple geographical locations that are then divided into different legal entities.

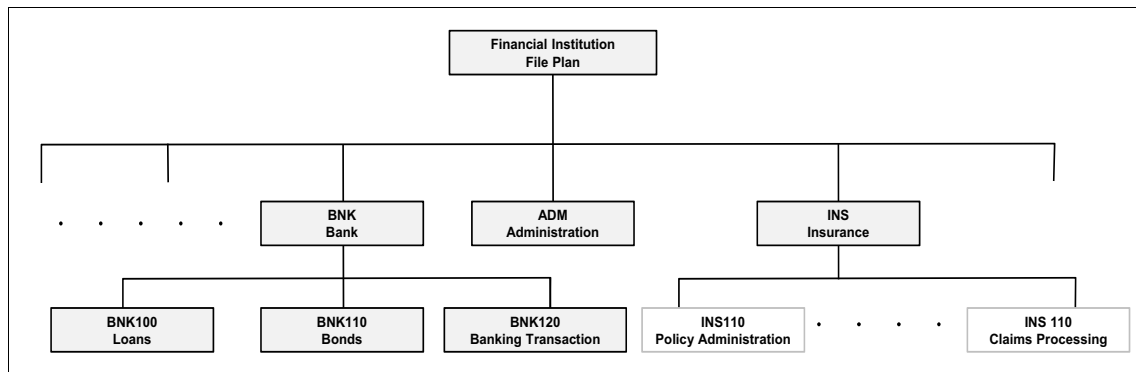


Figure 3-6 Partial file plan

A hierarchical file plan has containers (subcontainers and records) that comprise the file plan that represents categories of information. The highest level of folders is known as *root*. Subfolders are used to aggregate records based on business subjects. These categories are divided into narrower categories until a category is granular enough that it pertains to a relatively small subset of the spectrum of categories that an organization deals with on a daily basis. This level of granularity is where a specific retention rule can apply to this set of records, which we describe in 3.1, “Retention schedule” on page 66.

Users (Records Administrators) identify records as belonging to one or more of these narrowly defined categories. This granularity enables efficient location of records based on the categories of information to which they belong.

While designing a file plan, distinguish between electronic records and paper records, identify the location of these records and how and where they are created, and note any changes or migration of these records.

For information about how to create a file plan, see Chapter 14, “File plan case study” on page 295.

For information about how to apply retention rule to that file plan, see Chapter 14, “File plan case study” on page 295.

3.5 File plan in IBM Enterprise Records

To model an organization's file plan in Enterprise Records, it is important to understand the elements or objects that the software uses to build a file plan and how they interact with each other.

There are different ways of setting up file plans in Enterprise Records. The retention policy of records can be implemented in the Policy Management module (for example, IBM Global Retention Policy and Schedule Management). Master schedule and local schedules are based on certain criteria, for example, geographical locations can be syndicated to Enterprise Records.

Another way of implementing file plan is to enter it directly into Enterprise Records.

3.5.1 File plan elements

File plan elements in Enterprise Records 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.

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, all the nodes (such as employee data, corporate governance, banking, accounting, broker-dealer transactions) are modeled as record categories.

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 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 Enterprise Records.
Box	A box is a container for physical records and provides a mechanism to model physical entities that contain other physical items. It is analogous to a cardboard box in which physical records are actually stored. A box can contain only physical records. It can also contain physical folders and other box folders. A box does not use <i>volumes</i> . Quite often, an organization needs to merely 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 Enterprise Records does not have any elements that represent the individual items in the box.
Hybrid folder	A hybrid folder can contain both electronic and physical records and contains one or more volumes. There are no functional 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.

Note: Avoid mixing electronic and physical records in the same container. Typically, physical and electronic records have different processes associated with their disposition. By keeping them 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. However, 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 as shown in Figure 3-7 on page 86.

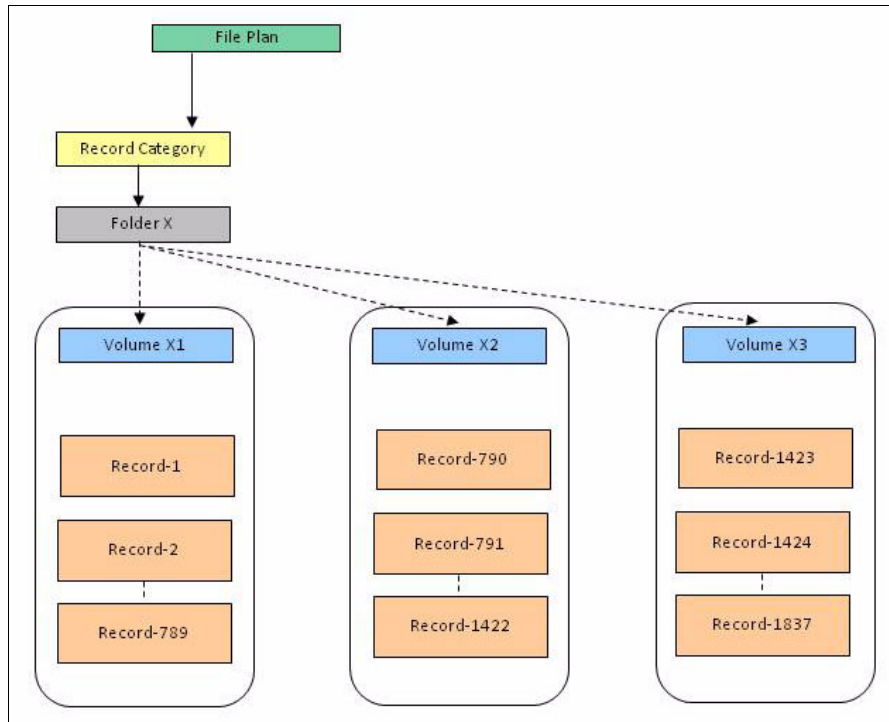


Figure 3-7 Records category, folder, and volume

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 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 must 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.

IBM Enterprise Records supports both electronic and physical records:

Electronic records An electronic record is a record that points to an electronic document stored in the IBM 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 Enterprise Records Manager (commonly referred to as a *marker*) is a pointer to a physical document or another object that exists in the organization, such as paper records, tape, or microfilm. 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 physically store it in only one box).

Figure 3-8 on page 88 illustrates the constraint relationship among file plan container entities and records.

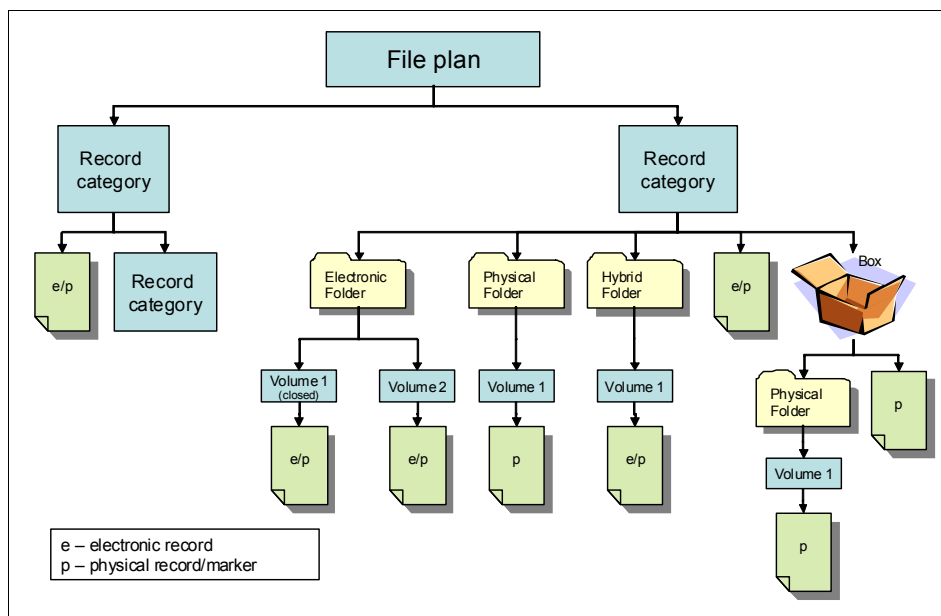


Figure 3-8 Constraint relationships among file plan elements

The file plan elements provide a flexible model in file plan design, but having this flexibility also raises the question of when to use folders and when not to use them. 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 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 the type of disposition.

3.5.2 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 subclassified 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 information about the attributes of containers and records, see the software documentation 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.

Enterprise Records 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 Enterprise Records application.

You can also mark containers as Permanent. There is no behavior associated with the Permanent Record Indicator property. That is, the property is informational in nature.

3.6 Case study: File plans in IBM Enterprise Records

Throughout this book, we use two case studies as examples to illustrate some of the options of implementing a file plan in Enterprise Records.

In our first example, the retention rules for a multinational financial institution take into consideration varying retention requirements for records.

In Enterprise Records, a basic disposition schedule is used to illustrate the record level aggregation that is described in Chapter 14, “File plan case study” on page 295. Record category is used to represent the implementation of functional level, such as accounting, broker-dealer transactions, corporate governance, insurance, and annuity.

Note: Basic disposition schedule is a high-performance schedule function that is intuitive to use.

Within each category, the records can be further broken down into smaller groupings. A basic schedule with immediate destroy without review by an administrator is shown in the case study. Records are destroyed after the review period expired for event-based client.

In our second example, a government agency, depicts single country and single jurisdiction. Region-specific retention requirements do not need to be considered, because retention requirements are governed at the national (federal) level. The government example has mostly case-based records groupings. We use Enterprise Records Advanced schedules to illustrate disposition in Chapter 16, “Advanced disposition case study” on page 327.

Note: An Advanced disposition schedule offers more capabilities and flexibility, but it is more complex and might affect performance.

In our government example, record categories such as Financial Management, Equipment and Store, and Compensation are represented at the functional level.

Within each category, records are broken up into further groupings for access and ownership purposes.

Disposition takes place at the folder level, such as Agreements. All of the records that pertain to the same contract or agreement are disposed of at the same time (seven years upon expiration of the agreement, for example). Records that are eligible for disposition are grouped for review.

In Chapter 16, “Advanced disposition case study” on page 327, the government case study illustrates *review and destroy* for one record series and *review* and either *destroy* or *transfer for client* for another record series.



Security

In this chapter, we describe how to control access to the records that are stored and managed in an IBM Enterprise Records repository. Security is a key component of a comprehensive records management solution, because it provides access control to the records in the system and ensures that records are destroyed only through a defensible process.

This chapter covers the following topics:

- ▶ Security model overview
- ▶ Records management roles and security
- ▶ Determining the security model
- ▶ Individual record security
- ▶ Security and record holds
- ▶ Limiting functional access
- ▶ Separating records into multiple repositories

4.1 Security model overview

One of the key features 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 a combination of security settings and how you choose to configure access based on the requirements of your records management solution.

By adjusting security settings and configuration, the Records Manager or Records Administrator decides who can access the content or the metadata associated with the content. 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, Enterprise Records 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 until dictated by retention policy, following the general best practice that records are only destroyed as part of a well-defined, defensible disposition process. Under certain circumstances, authorized users, such as Records Administrators or Records Managers, can delete or undeclare records to correct mistakes or errors in the declaration process. However, the deletion of all records is fully auditable.

Enterprise Records relies on the powerful capabilities of the IBM Content Platform Engine security model. Implementing a robust security schema involves understanding the various options available, such as how the file plan tree can provide a primary structure for imposing inherited security on the records that are declared in the file plan. It is also important to understand the additional features and capabilities of the IBM Content Platform Engine security model that can be applied to records, such as security proxies, inherited security, and marking sets. Implementing a successful records security schema requires the following knowledge and actions:

- ▶ Understanding the best practices for access to business records
- ▶ Understanding the predefined Enterprise Records security roles and how you can use them
- ▶ Identifying typical 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)

Note: Defining the appropriate security schema is a key aspect of designing a comprehensive records management solution.

A security schema might include these choices:

- ▶ Adding simple departmental or business unit separation to the file plan
- ▶ Using the partial security proxy type to allow previously configured access settings to be retained while extending access for purposes of managing and disposing of records
- ▶ Using marking sets to impose additional access restrictions on individual records
- ▶ Using direct security or custom proxy objects to apply varied access control to individual records

As you learn about the security features that we describe in the remainder of this chapter, remember that the Content Platform Engine offers more options and capabilities for modeling and controlling security than those covered in this book. For more detailed information about the various aspects of the Content Platform Engine security model, such as how inherited security works, how to add and configure marking sets, and how to use field masking to control access for specific properties, see FileNet P8 security documentation in IBM Knowledge Center:

<http://ibm.co/1dT8Smi>

4.2 Records management roles and security

When designing a records management system, whether it is manual or electronic, there are a variety of 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 define both the actions that a user is permitted to do and restricted from performing.

Enterprise Records has a set of predefined security roles that provide a framework and starting point for defining your records management security. Each of these roles has been assigned specific access levels to the various functions and features available in the system. The first step in creating a security model is to understand the access levels that are defined for each of the predefined roles. This helps you 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

Four standard roles are defined by the Base data model of Enterprise Records:

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

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

Preferable practice: Avoid mapping individual users to Enterprise Records security 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 affecting the configuration in Enterprise Records.

There is variation on the predefined roles with the other data models. For example, the DoD Classified data model implements an additional role called the Classification Guide Administrator.

It is a common practice to further differentiate the standard security 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 have access only to the records that belong to that department. We describe this topic in more detail later in this chapter.

Now, we focus on the four standard roles that are already defined in the software 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 implementing.

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 rather than the business unit or records management team. This user helps to manage the 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 and in the file plan object store (FPOS).

These tasks are commonly associated with the Records Administrator role:

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

Many of the tasks the Records Administrator must perform are done through administrative and configuration tools outside of the Enterprise Records desktop interface.

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:

- ▶ Define the structure of the file plan by determining the appropriate record categories that will establish the classification scheme for records.
- ▶ Define and update disposition schedules.
- ▶ Allocate disposition schedules to record 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 of records.

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 record folders within a given category, if folders are being used.
- ▶ Declare records or organize records for automated declaration.
- ▶ Update record properties or move records if needed.
- ▶ 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.
- ▶ Create new records by adding content to the repository with the appropriate metadata that will determine where and how the record will be declared.
- ▶ Identify electronic documents for declaration and participate in declaring records through a well-defined business process.
- ▶ Store and retrieve physical records.

In many cases, the Records User might not require access to the Enterprise Records desktop interface at all, but will instead rely on search and browse capabilities provided by their standard IBM Content Navigator desktop or some other application interface.

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 when setting up a new FPOS object store, which typically happens for the first time during initial software installation. The security groups are established on the FPOS object store during object store creation and mapped to the security roles as part of the initial object store configuration process.

Knowing which security groups to map to each of the Enterprise Records security roles is important. The most flexible approach is to create one master security group for each of the Enterprise Records 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 Enterprise Records.

Establishing master security groups

In Table 4-1 on page 100, we illustrate four security groups where we use a naming convention that associates the groups with Enterprise Records 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 will probably add other groups that have individual users as members. The naming convention that we chose here is merely an example. Most organizations establish their own standards and policies for naming security groups.

Table 4-1 Mapping security roles to master security groups

Security roles	Example master security groups
Records Administrator	IER_RecordsAdminG
Records Manager	IER_RecordsManagerG
Records Privileged User	IER_PrivilegedUserG
Records User	IER_RecordsUserG

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

Figure 4-1 shows an example security mapping between roles and master groups for an example FPOS object store. This information can be accessed through the Enterprise Records desktop Administration view by selecting the FPOS repository and viewing the Security Script tab. This mapping is established during the initial configuration of the FPOS object store. Although the mapping can be changed by adjusting which groups are used for each role, the change in mapping does not apply to objects already created. So, it is best to establish this mapping before building out the file plan configuration.

RB Record Catalog 1

Repository: RB Record Catalog 1

General Display Columns Display Properties System Properties Security Script

This task may take several minutes to complete. Security updates apply to new object instances only. If you have custom security settings that are placed on records management objects and you run the security script, the settings might be overwritten. Administration actions like upgrading IBM Enterprise Records require you to be an original object store administrator and a records administrator. You must be aware of this requirement when you assign members to the Records Administrator role.

Restore Defaults Run Security Script

Records Administrator: RMAAdminG x Add...

Records Manager: RMMManagerG x Add...

Records Privileged User: RMPPrivilegedUserG x Add...

Records User: RMUserG x Add...

Figure 4-1 Example security mapping between security roles and LDAP groups

Using the predefined security configuration

When setting up the initial security configuration as just described with one master security group for each of the record management 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 security roles by assigning existing organizational groups to the master Enterprise Records 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 Enterprise Records. For example, you might already have a group called `Example_RecordsCenterStaff`. This group can be added as a member of `IER_RecordsManagerG` to give all the users who are members of the Records Center full Records Manager access.

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

In Table 4-2 on page 102, we provide an example that shows how you can manage existing security groups in your organization by establishing group membership rather than changing the direct mapping of security roles to the master security groups. In this particular example, we assume that all members of the `Example_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 `Example_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 IBM Enterprise Records security

Master Records Manager security groups	Existing directory services groups
IER_RecordsAdminG	Example_IT_P8Admins
IER_RecordsManagerG	Example_RecordsCenterStaff
IER_PrivilegedUserG	Example_Dept_1_Coordinators Example_Dept_2_Coordinators Example_Dept_3_Coordinators
IER_RecordsUserG	Example_Dept_1_Users Example_Dept_2_Users Example_Dept_3_Users

groups

Preferable practice: Rather than assigning individual users to the master 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 Enterprise Records roles and can be assigned to the master Enterprise Records 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 Content Platform Engine 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. When using nested groups, always consider the performance impact that might arise if there are a large number of nested groups.

4.3 Determining the security model

Enterprise Records offers a variety of options for configuring the security schema that will determine how access is controlled. The options available will depend on which security proxy type is selected.

4.3.1 Security proxy types

Each ROS can be configured to use only one of three different security proxy types that will determine how access control is applied to each document that is declared a record. This setting applies to the entire ROS object store and, therefore, applies to all documents in a given object store that are declared as records. By default, each ROS is initially configured for full proxy, so this setting must be changed by the Records Administrator if one of the other proxy types are required.

The following security proxy types are available:

- ▶ *Full proxy*: the file plan security configuration overrides the original security settings applied to the document by its home objects store (ROS). The access control is determined by the security configuration established by the file plan and the records configuration in Enterprise Records.
- ▶ *Partial (Inherited) proxy*: the file plan security is inherited and added to the access control defined for each document in its home object store, thereby allowing the existing document security model to remain in place while enabling additional records management access controls to be applied
- ▶ *No proxy*: the record declaration process does not affect the access control of any documents that are declared records, thereby allowing the existing security model to remain unchanged after a record is declared, except for preventing deletion. Just as with the other proxy types, each record will be locked down so it cannot be deleted directly by users, even if those users have delete permission on the original document.

For detailed information about how to configure the ROS object store security proxy type, see the IBM Technote:

<http://www.ibm.com/support/docview.wss?uid=swg21591738>

Each option has its tradeoffs. The following considerations can be used as guidelines to help determine the best option for a given solution. In general, the partial proxy type provides the most flexibility and aligns with many common use cases.

- ▶ Full proxy is useful when:
 - There is no existing security model for a new implementation and all documents will be declared as records.
 - You want the security to change when a record is declared and you want that access to be determined by the file plan.
 - Your file plan structure provides separate containers (categories or folders) to designate who can access records.

- After a record is declared, any further adjustments to security would be controlled by the records management staff.
- ▶ Partial (Inherited) proxy is useful when:
 - There is an existing security model already in place with the access control that you want, so you only need to enable records management access after declaration and documents are locked down to prevent delete.
 - The file plan structure does not align with how you want to assign permissions to records.
 - You require security settings to vary on individual records where you can control this on the original documents (such as applying markings or using custom security proxy objects).
 - You have the need to adjust access on individual records independently of records management processes after a record is declared.
- ▶ No proxy is useful when:
 - You do not want to change your existing security model and you can easily add the appropriate access for records management functions to the ROS object store.
 - You have complex access control requirements that are best modeled directly on the home object store (ROS).

By using either the full or partial security proxy options, the file plan not only provides a structure for assigning disposition schedules, the file plan can also provide a structure for assigning access to the records in the file plan, which in turn controls access to the associated electronic content. The file plan uses the IBM Content Platform Engine 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.

4.3.2 Containers as security parents

Enterprise Records establishes *record containers* (the categories and folders into which records are filed) as the security parents for the records that are contained. Therefore, 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, Enterprise Records was designed to use the security parent mechanism and inherited security to allow 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, see the IBM Content Platform Engine documentation in IBM Knowledge Center:

<http://ibm.co/1dT8Smi>

4.3.3 Controlling security by full 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 full security proxy for the electronic document that it controls in the repository.

The inheritance that is described in the following sections will work with either the full proxy or the partial proxy security types to control access.

4.3.4 Relating file plan structure to access control

One of the easiest ways to provide simple access separation to records is to construct the Enterprise Records file plan to have separate containers for the records belonging to different parts of the organization, if such separation is needed. For a given record series, you can create separate categories for each department or business unit that uses that record series. This approach not only makes it easy to apply the appropriate access to the records, but it also helps to organize the records for review and disposition based on ownership or responsibility. However, the primary file plan structure will still be determined by how records are organized for retention.

The primary file plan structure rarely matches the way that you need to separate records for ownership and access control. Typically, an organization can have hundreds of departments or units, each of which might use any area of the primary 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 series or record types that are used by a wide spectrum of the organization. Therefore, these lower-level primary categories cannot be mapped to specific individual departments without further breakdown. 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 many different departments in the 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 many of these cases, the primary file plan structure will not provide a good mechanism for allowing independent, more granular access, such as the access that is required for individual departments.

Figure 4-2 on page 107 illustrates several important concepts for setting up access control with the file plan structure.

First, for records management and administration, you can set the access at higher levels in the file plan so that it is inherited to all levels under them. This allows service accounts and records management users with global access requirements to view and process any record in the system.

Second, you can assign access to specific departmental groups at lower levels of the file plan that will limit their access to only the records contained under that branch of the file plan structure. As the example shows, the category for Employee Identity records (record series EMP110) could contain all records from all departments in the organization, but the only users who should be able to view this data are the HR Department Users. However, you might not want all Human Resource (HR) Department Users to view all Compensation records (record series EMP100). So, by adding the HR Department Users group to the appropriate categories, we can prevent those users from accessing Compensation records, while allowing them to access all Employee Identity records. This simple example of separation of access happens to align with the primary file plan structure.

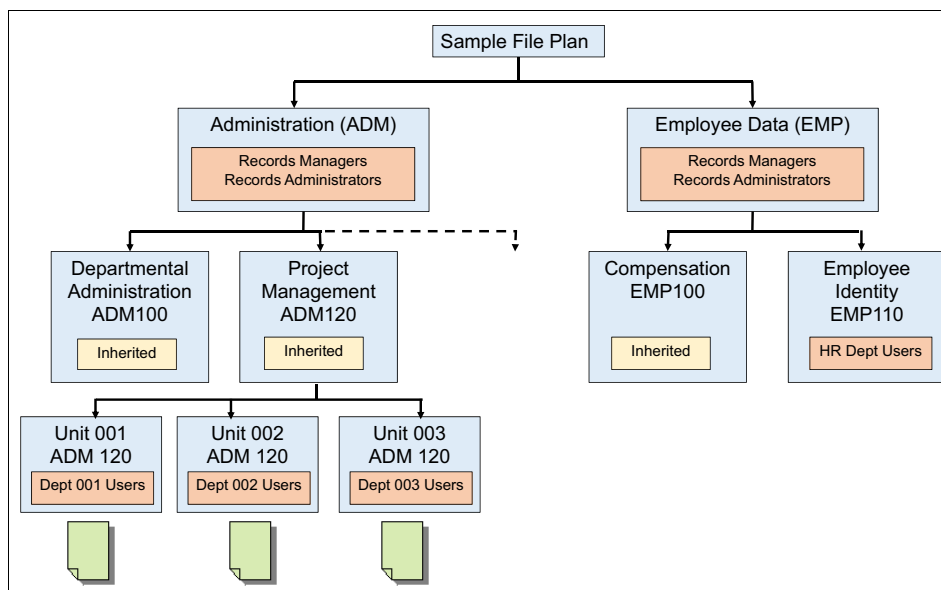


Figure 4-2 Assigning department-level security to categories in the file plan

As a different example, take Project Management records (record series ADM120). Any department that manages projects will need to keep track of their own project management records and they do not want people in other departments to be able to view their records. So the example in Figure 4-2 shows how we can create a separate category for each organization unit that needs to store Project Management records and assign the appropriate departmental groups as needed to limit access.

Preferable 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 described here to implement a robust security schema to meet your business requirements.

4.3.5 Access control that differs from the file plan structure

The examples described in 4.3.4, “Relating file plan structure to access control” on page 105 might not provide the granularity required for certain use cases. There are a variety of techniques and approaches that can be used within the IBM Content Platform Engine security model to provide the access control that might be required.

In the next section, we describe two basic techniques for applying access control independently of the file plan structure:

- ▶ Marking sets
- ▶ Direct security

In addition to these two basic techniques, there are also more advanced techniques, such as applying custom security proxy objects to individual records. In these cases, using the partial security proxy type for the Enterprise Records security configuration will typically offer the most flexibility to support the combination of individual record access and records management level global access.

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:

- ▶ Using marking sets to apply a marking value to individual records
- ▶ 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 Unclassified, Confidential, Secret, and Top Secret. 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 to access that record. List markings can be 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 consists 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 the appropriate marking value for a department is assigned to a record, 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 if 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	Dept_1_Users Dept_1_Coordinators IER_RecordsManagerG IER_RecordsAdminG	View only View and update Full control Full control
Dept 2	Dept_2_Users Dept_2_Coordinators IER_RecordsManagerG IER_RecordsAdminG	View only View and update Full control Full control
Dept 3	Dept_3_Users Dept_3_Coordinators IER_RecordsManagerG IER_RecordsAdminG	View only View and update Full control Full control

Another aspect of the example that we provide here shows that both IER_RecordsManagerG and IER_RecordsAdminG 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. Therefore, 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 109, you can then apply marking values to individual records in the file plan, to the original documents in the content repository, or to both depending on which security proxy type you are using and how the users are planning to access the records.

Figure 4-3 on page 111 illustrates three individual records contained in the Compensation 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 the Dept 1 marking and one is assigned 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. In addition, inherited security still gives us the ability to restrict the HR Department Users to from accessing any Compensation records by implicitly excluding that group.

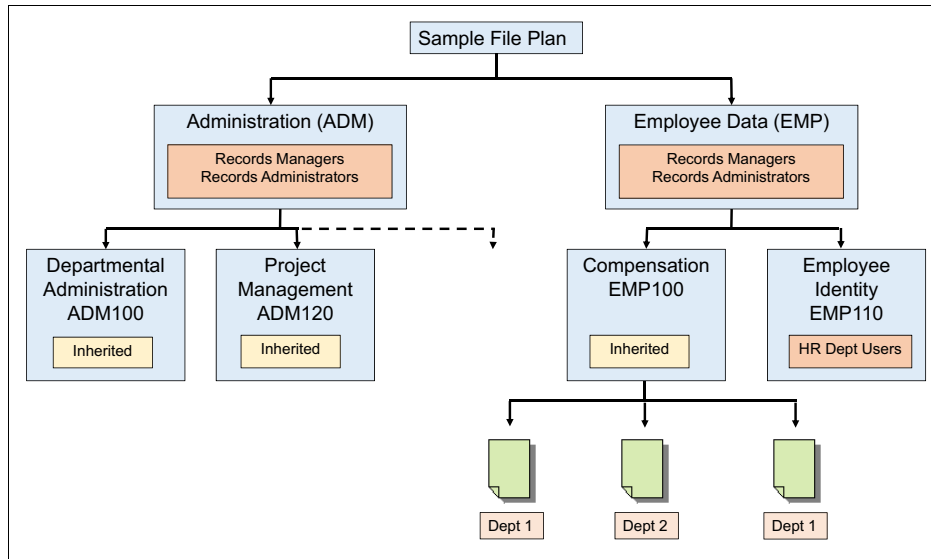


Figure 4-3 Assigning markings to individual records to control access

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 an automated process 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 Enterprise Records 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 rather than simply marking each record with a marking value.

Figure 4-4 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 rather than simply applying a marking value to each record. Notice that the records will still inherit security from the file plan. This allows both Records Managers and Records Administrators full control and full access to all parts of the file plan. In addition, HR department users will be restricted from Compensation records because they are not explicitly assigned to that branch of the tree. However, users in any of the other departmental groups will have access to only the individual records where those permissions apply.

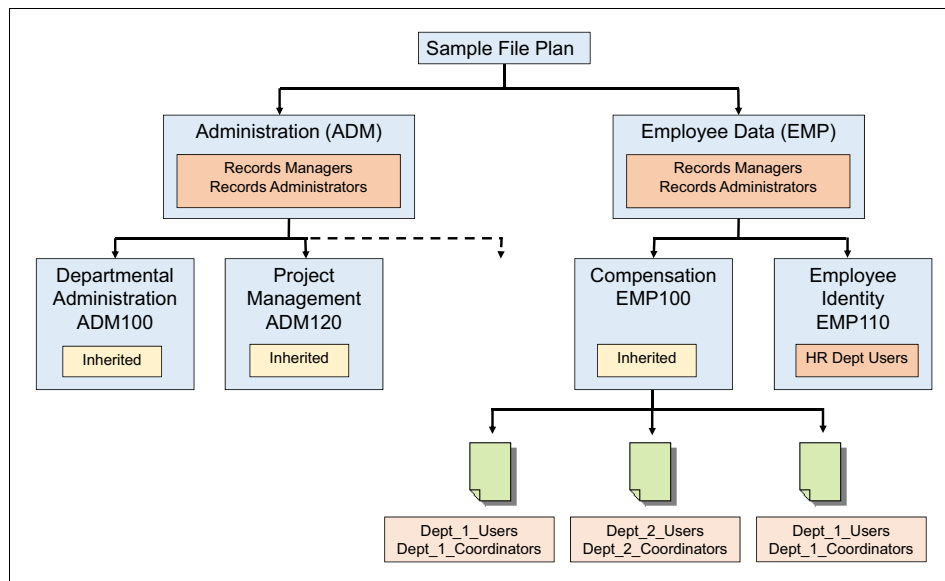


Figure 4-4 Assigning direct security permissions to individual records

4.4.3 Comparing approaches

Both marking sets and direct security allow for a flexible security schema, which enables 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 administrative overhead of assigning a 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 need to modify the security permissions related to the group abstraction that you are using, you will probably have to modify the permissions on each record. For example, imagine that you want the Dept_1_Users to now have update permissions in addition to viewing content. With marking sets, you make that adjustment in one place on the marking set. With direct security, you must make an adjustment to possibly hundreds or thousands of records.

Although marking sets provide this powerful abstraction, there is additional effort 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 to provide adequate access control. If 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 additional effort 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. Typically, it should be avoided unless it is managed by a custom application.

4.5 Security and record holds

No matter what access level you have, when a record is on hold, the Enterprise Records system prevents any user from deleting or destroying that record. Even if you are a system administrator with full access control, the Enterprise Records 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 that container.

For more information about record holds, see Chapter 8, “Holds and preservation” on page 213.

4.6 Limiting functional access

The previous sections of this chapter focused on how to control access to the records in the repository by applying permissions in various ways. With Enterprise Records, we can also control access to various functions in the user interface by configuring one or more Enterprise Records desktops to limit user access to the desktop and to limit what features and functions appear in a given desktop. Editing the desktop features and layout is an administrative task that requires the IBM Content Navigator administration desktop.

4.6.1 Limiting access to a desktop

When first setting up an Enterprise Records environment, we typically start with one general Enterprise Records desktop that exposes all features and functions to any user who can authenticate to the specified file plan object store (FPOS). However, in a fully configured solution, we will probably want to limit which users and groups have access to such a desktop.

In Figure 4-5 on page 115, we show an example where we have chosen to limit access to the particular desktop to only Records Administrators and Records Managers. We want to do this because we plan to set up separate desktops for other users, where we will limit which functions and menus items appear, but we want Records Administrators and Records Managers to have full access to all functions and features.

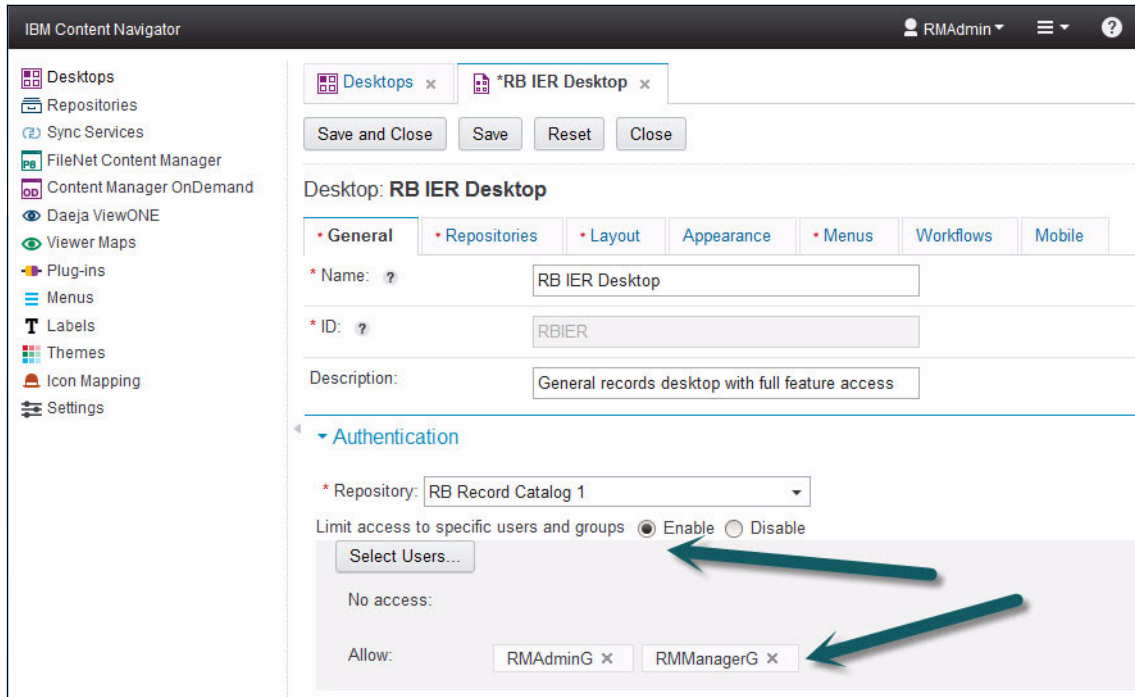


Figure 4-5 Limiting desktop access to specific users and groups

When any user attempts to connect to the desktop, the desktop will authenticate access for that user based on the repository indicated and, if configured (enabled), will limit access to the users and groups listed. In this example, if we attempted to connect to this desktop as any user other than a Records Administrator or Records Manager, we see the message in Figure 4-6 on page 116.

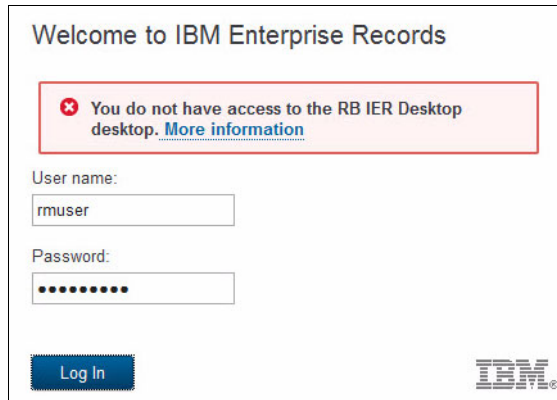


Figure 4-6 User does not have access to the restricted desktop

4.6.2 Limiting access to features

Many Enterprise Records configurations use a limited set of the software's features. In such cases, it might be desirable to define a separate desktop to expose only the features that particular group of users might need and remove other features from the interface. For example, in our sample use case for a financial institution where we are managing only electronic records and using only basic schedules as record containers in the file plan, we could configure a separate desktop for Records Managers that would expose only the main features needed.

As shown in Figure 4-7 on page 117, for this use case, we configure a separate Enterprise Records desktop for Records Managers where we remove the Physical Items view and add the Work view.

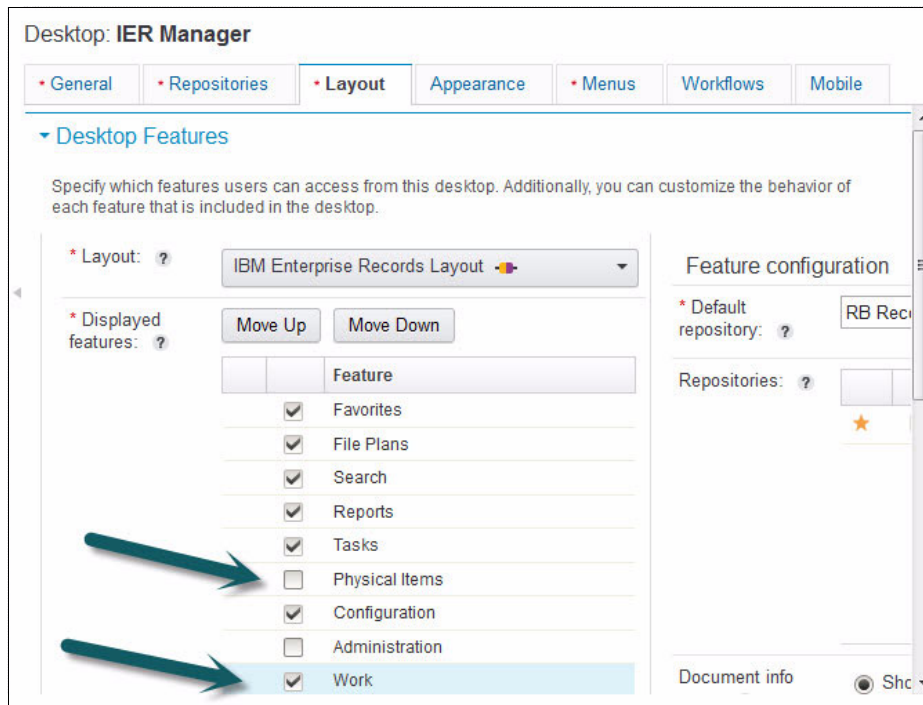


Figure 4-7 Configuring the desktop features to display

When setting up a new desktop, the Work view is not selected by default, so it must be selected to enable access to the workflow features.

Note: The Administration view will be displayed only for users who have administrative permissions on the FPOS object store that is selected for desktop authentication. Among all of the Enterprise Records master groups, only the Records Administrators are typically object store administrators. So even if you enabled this feature for a Records Manager desktop, the Records Manager users would not see the view.

The ability to configure which features are displayed for a desktop implicitly allows access to be limited to certain functions by limiting which users can access a specified desktop.

4.6.3 Limiting access to specific functions within a view

IBM Content Navigator enables menus to be configured to limit which functions are available. You can configure and assign alternate menus for a specific desktop to control which functions appear in specific menus.

4.7 Separating records into multiple repositories

For an enterprise-wide, global deployment, it might be desirable to separate records based on jurisdiction. Even though this can be accomplished when storing all records in the same object store, it is a typical practice to manage separate content, and the corresponding record information, by jurisdiction. How that separation is managed is dictated by the requirements of the solution. Records could be separated by country, by region, or by any other logical jurisdictional distinction, such as separation by primary line of business. This sort of separation is primarily guided by the strict access and storage policies imposed by certain countries, along with the variation in retention policy details that exist between countries. Because all of the record information for a specified repository is kept in a single database, the best way to ensure that the information is stored and protected accordingly by a local jurisdiction is to separate that information by object store.

IBM Enterprise Records and the IBM Content Platform Engine support the scalability that is afforded by using multiple object stores to manage records across the enterprise. The combination of using a robust security schema with support for access to multiple object stores from the Enterprise Records desktop and the ability to configure separate Enterprise Records desktops means that you can configure a solution for the most demanding access control requirements.



Records capture, creation, and retrieval

This chapter describes how to configure IBM Enterprise Records software to support records creation and capture, where record creation and capture are understood to be the process involved in creating the record and then capturing the record in IBM Enterprise Records. We review various mechanisms available to import and process, or *ingest*, content, describe how these mechanisms can be combined with a variety of record declaration mechanisms, and cover the benefits and constraints that come with the common choices available.

We cover the following topics in this chapter:

- ▶ Why automation is the goal
- ▶ Record capture
- ▶ Manual record creation and capture
- ▶ Performance considerations

5.1 Why automation is the goal

To maintain a holistic and accurate enterprise records management system, it is essential to understand how new records are ingested and declared. Enterprise Records can use the full capabilities of the IBM Enterprise Content Management portfolio, including bulk import tools, such as the IBM Content Collector products, which provide a myriad of options for record creation and capture. Although IBM Enterprise Records supports both automated and manual options, most customers prefer automated or non-discretionary record capture as the mechanism to ensure that records are created and captured in a consistent and reliable manner. Usually, the mechanisms used for record creation and capture are guided by requirements including volume, frequency, and consistency or variation, although the cost associated with automation might influence a decision to support manual record creation and capture.

The following are some examples of common use cases for record creation and capture which, tend to lend themselves to an automated record creation and capture process based on volume, frequency, and consistency:

- ▶ The enterprise creates the record, in large volumes periodically, and the record is usually of a particular type. Examples:
 - A financial services organization producing customer banking statements
 - A government entity producing outbound appointment notifications
 - A utility provider producing billing notifications
- ▶ The enterprise receives records from customers or entities with sufficient frequency, where the records received are consistent in type or nature. Examples:
 - Invoices and credit notes
 - Delivery dockets
 - Application forms
 - Taxation documents
- ▶ The enterprise has an approval process (so there is a frequency of occurrence) which generates correspondence with customers or entities, such that when the item is approved or released, it is captured as a record. Examples:
 - Letters of offer
 - Contracts or agreements
 - Financial arrangements
 - Customer complaints

- ▶ The enterprise generates documentation with sufficient frequency to support a document lifecycle process which automates the record creation and capture process. Examples:
 - Board meetings and papers
 - Organization policies or procedures
 - Training documentation.

Typically, where customers decide to use manual or discretionary methods for record creation and capture, there is lower volume, less frequency of occurrence, or greater variance in the types of records captured and created, which results in a greater level of user involvement with the record creation and capture process. The following are some example of common use cases for manual record creation and capture within enterprises or organizations:

- ▶ Unique or complex contracts with third-party entities
- ▶ Legal documents
- ▶ Human resources documentation
- ▶ Industrial relations
- ▶ Events and ceremonies

These are just a few examples of the types of records the enterprise creates and captures into IBM Enterprise Records. The remainder of this chapter covers both automated and manual methods of record creation and capture.

IBM does not make any recommendations as to whether an enterprise uses automated or manual methods for record declaration, although our customers who alleviate users of the burden of manual declaration tend to find records are captured and created with greater consistency and reliability.

5.1.1 Successfully automating record creation and capture

Successfully automating record creation and capture for an enterprise requires the use of methods to automatically archive records, and the ability to automatically identify and classify the record accordingly, as generally, not all records archived will be of a single type of record in the enterprise. When the methods for auto-archiving are configured and defined, and the classification tools have set up and configured, the enterprise can benefit from automated record creation and capture.

Auto-archiving

Successfully automating archiving in an enterprise requires understanding the enterprise's records, and how or where they exist before capture into IBM Enterprise Records. Based on our experience, many customers have policies to define what a record is, under what conditions a document becomes a record, where it can be held in the organization, and when and how it should be captured into the records management system.

IBM understands records might arrive into the enterprise through many mechanisms including these examples:

- ▶ Hardcopy documents
- ▶ Faxes
- ▶ Multifunction device and printer (MFDs or MFPs)
- ▶ Emails
- ▶ Files on a file system
- ▶ Files in a system of engagement such as IBM Connections or Microsoft SharePoint
- ▶ Office productivity tools, such as Microsoft Office
- ▶ Document Management Systems
- ▶ Enterprise resource planning (ERP) systems such as SAP
- ▶ From mobile devices

By noting the diversity of channels which might be used in an enterprise to receive or generate records, it becomes apparent that multiple methods for auto-archiving records might be required by an enterprise. These tools often remove the requirement for users to nominate when an item should be auto-archived, but are flexible enough to support use cases which provide for a level of automation which might be based on some user discretion.

The following are some examples of auto-archiving to IBM Enterprise Records:

- ▶ An enterprise might state that not all emails received are records, but those emails which are records must be flagged by a user, or moved to a location to allow them to be automatically archived into IBM Enterprise Records.
- ▶ Alternatively, an enterprise might state that all emails received in the enterprise or in a specific mailbox or set of mailboxes must automatically be archived into IBM Enterprise records in a non-discretionary manner.

IBM supports both of these use case examples, automatically archiving emails into IBM Enterprise Records through use of IBM Content Collector for Email.

- An enterprise might stipulate items stored on a file system for longer than six months should be captured and managed as records.

IBM supports auto-archiving of data on the file system by referring to data such as date created, or date last modified or accessed into IBM Enterprise Records through use of either IBM StoredIQ Policy Assessment and Compliance or IBM Content Collector for File Systems.

For further information about IBM supported auto-archiving tools, see *IBM Value-Based Archiving* website at the following web address:

<http://www.ibm.com/software/products/en/value-based-archiving>

Auto-classifying

Automatically classifying records into IBM Enterprise Records requires the ability to identify what the record is and then to appropriately classify the item according to the File Plan in IBM Enterprise Records. IBM Content Classification can provide this ability by analyzing the full text of documents and emails and applying rules that automate classification decisions.

Classification can also be used to determine whether a content item must be classified or not. By filtering out email about lunch appointments or documents that do not hold any business value, for example, you can reduce costs and ensure that only documents that must be retained are classified and archived.

Embedded with natural language processing and semantic analysis capabilities, IBM Content Classification determines the true intent of words and then uses that knowledge to automate decision making. Unlike other classification systems that are based on rules only, IBM Content Classification combines rules and contextual analysis to incorporate real-time learning that adapts to changing business needs. As a result, classification becomes even more accurate over time.

A content classification corpus

To automate the classification of records into IBM Enterprise Records, it is necessary to train IBM Content Classification with your enterprises records and file plan. You do this by providing a predefined corpus of your enterprises records and your enterprise file plan. IBM Content Classification reads the records in each category of the file plan, and creates the association between the category and records which will provide the most accurate results. After the results are tested and refined, a Content Classification knowledge base is created, which can be to identify new records and classify the records appropriately.

It is important to use the reports generated by IBM Content Classification to ensure an optimal result is achievable by the knowledge base. The reports will identify overlapping categories, where the categories and records are too similar and will result in sub-optimal auto-classification results.

Using the classification knowledge base

After the IBM Content Classification knowledge base is published and released, it can be configured to be used within the following:

- ▶ IBM Content Collector: If you use IBM Content Collector, content can be classified by IBM Content Classification when it is captured by IBM Content Collector. IBM Content Classification returns information that helps IBM Content Collector determine the appropriate action to take.
- ▶ IBM Classification Center: You can use the Classification Center to manage the classification of content that is stored in the repository. You use this web application to select the content to be classified, configure classification options (such as the decision plan to use and various run time preferences), monitor classification activity, and view the classification results. If necessary, you can also use the Classification Center to reclassify documents if you determine that different knowledge base categories or decision plan actions are more applicable.
- ▶ Custom applications: IBM Content Classification runs a set of server-side processes on one or more servers, and provides several client libraries for remote access that are designed for various development environments, including:
 - C/C++ development
 - Java development
 - Visual Basic (ASP) development
 - .NET development

Your choice of a client library is based on your application programming language and development environment. You might use more than one client library in the same application. For example, you can have server-side components written in C++ that interact with the system by using the C client. At the same time, Microsoft Internet Information Server (IIS) server-side scripts in ASP (VBScript) can communicate with the system by using the COM client, which is easily accessible from VBScript.

The software development kit (SDK) includes everything that is needed to develop applications for IBM Content Classification: client libraries, online reference guides, and sample application

Auto-creation

Application of methods for auto-archiving and auto-classification together provide a proven means for automating the creation and capture of records into IBM Enterprise Records.

5.1.2 The complexities of manual record creation and capture

Those enterprises that choose to use manual processes for the creation and capture of records into their enterprise records system face several issues in doing so including the following problems:

- ▶ *Resourcing.* Manually creating and capturing records is resource-intensive. For example, if it takes one minute to create and capture a record, a user dedicated to this task could create and capture only 500 records in a standard 8-hour working day.
- ▶ *Inconsistencies.* A reliance on manual record creation and capture is subject to user interpretation and might lead to inconsistency in classification of resources across a tool.
- ▶ *Human error.* Manual record creation and capture are subject to more human errors in both the consistent application of metadata and the consistent application of classification.

The complexities of manual declaration are mentioned at this point to ensure enterprises using IBM Enterprise Records are aware that automated options for records creation and capture enhance the consistency of declaration, reduce the burden on the users, and can assist in maintaining the accuracy and consistency of records captured.

Automated techniques might augment manual record creation and capture, to reduce the impact or burden on users, by allowing manual triggers to occur in the record creation and capture process, which then use elements of the automated record creation and capture process to complete record capture.

5.1.3 Overview of content ingestion and declaration

To make the best choices in designing and implementing an IBM Enterprise Records system, it is important to understand the difference between electronic content ingestion (record creation) and record declaration (capture).

Difference between ingestion and declaration

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

Declaration is the process of classifying (capturing) content into an IBM Enterprise Records file plan such that the content is under IBM Enterprise Records 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 Enterprise Records, *records* are only those electronic documents that have been explicitly declared as records in an IBM Enterprise Records file plan. If a document is added to an IBM FileNet Content Foundation 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 Content Foundation 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.

Choosing the correct declaration method

IBM Enterprise Records offers a great 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 Enterprise Records 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 Enterprise Records supports both immediate and delayed declaration as illustrated in Figure 5-1 on page 127.

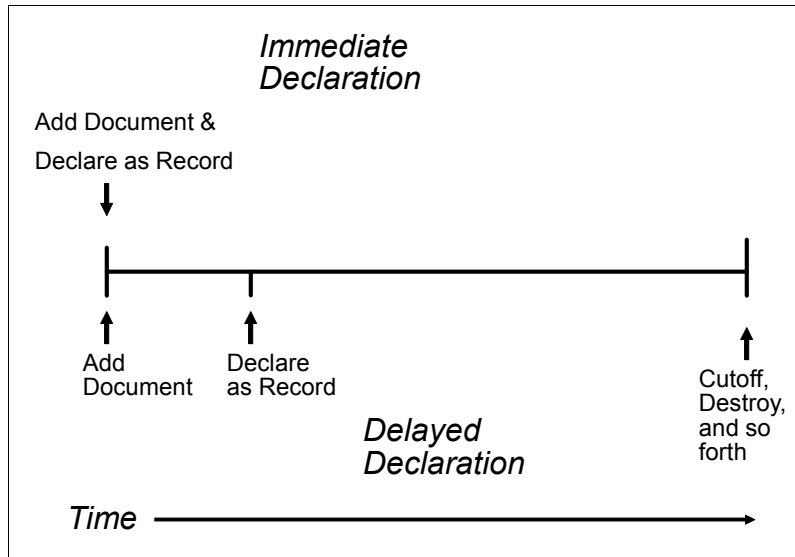


Figure 5-1 Beginning of the record lifecycle for immediate and delayed declaration

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.

How user roles Impact declaration choices

For purposes of ingestion and declaration, the enterprise must decide on which user roles have permission to declare records. Many enterprises differentiate between users who have view-only access to records and users who have permission to declare new records. For example, a user who is the author of a document is not necessarily the same person which 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 Record capture

Understanding the data to be stored in your IBM Enterprise Records repository, along with how, when, and who creates the data, is key to a successful records management deployment. The considerations vary dependent on whether your enterprise document and records repository is to be used to retain a subset of your enterprises documents, such as client documentation or corporate legal documentation, or whether it will store all records for the entire enterprise including all line-of-business records, and administrative records. Figure 5-2 illustrates this data dilemma.

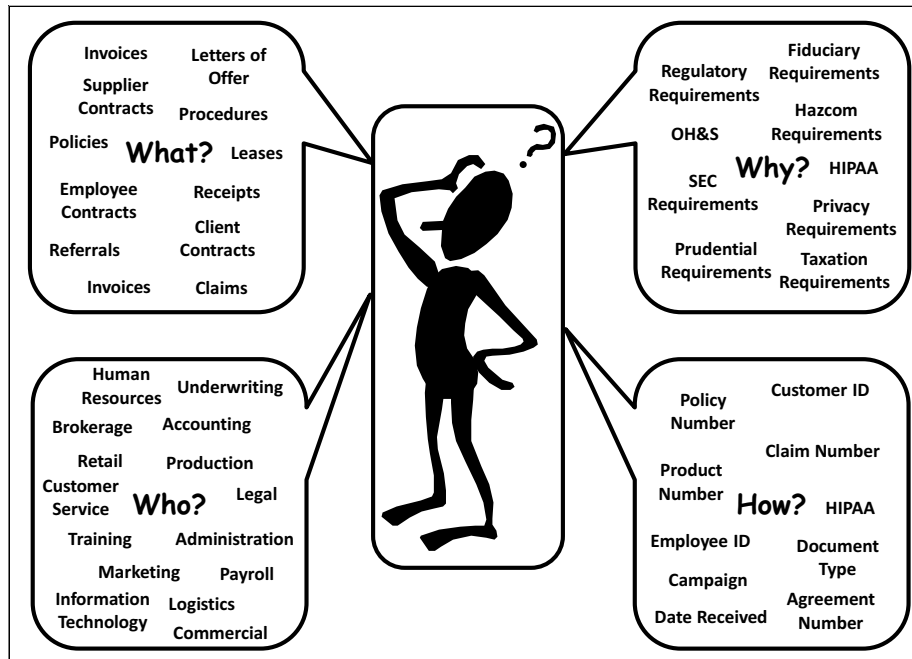


Figure 5-2 The data dilemma

When you are able to make an informed decision on the type of information to be stored in the repository, how it will get there, and how it will be identified, you must design and configure your document and record classes with the appropriate metadata. As described in Chapter 2, “IBM Enterprise Records system and architecture” on page 35, when a new record is added to IBM Enterprise Records, the content is actually added initially to the record-enabled object store (ROS) and for records management purposes is filed into the file plan object store (FPOS). Given most users will only interact with the ROS, there needs to be sufficient metadata to allow you to identify and retrieve the content from the ROS. The content identification metadata stored about the item in the

FPOS is required for primarily for records management and compliance purposes. It is the combination of document class metadata and record class metadata that, when combined, provide the complete metadata profile of the record, as illustrated in Figure 5-3.

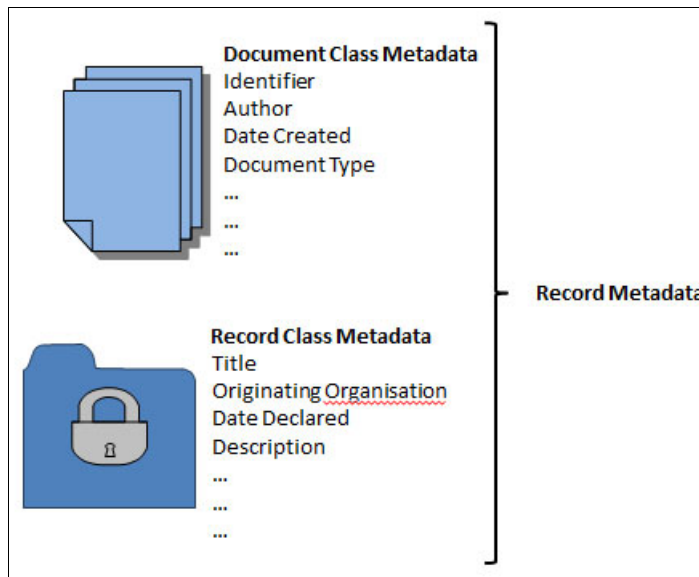


Figure 5-3 Complete record metadata profile

Note: A record metadata profile is the combined metadata of the document class and the record class.

After you understand the origins of the information to be added to IBM Enterprise Records and how it will be identified, the underpinning platform must be prepared for use, to support the business requirements for documents and records.

5.2.1 Document and record classes

After your platform is installed and configured for operation, 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 about

which classes you enable for record declaration. Figure 5-4 shows a customer-defined document class (starting with ILG) that is records-enabled. This document class is defined in the RB ROS 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.

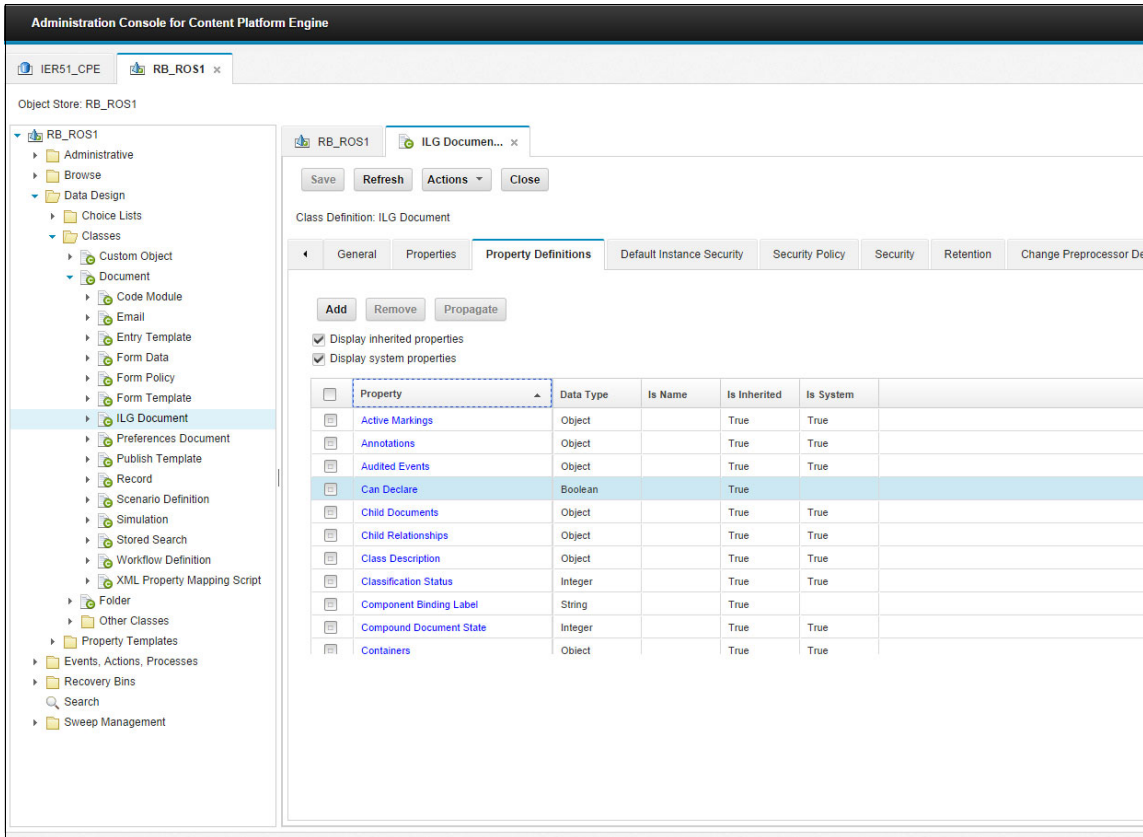


Figure 5-4 A customer-defined document class

When you define a document class, you also need to determine the properties associated with the class. The choice of properties is primarily guided by business requirements for search and retrieval, and 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.

Note: 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.

Record classes

After you have identified the records-enabled document classes for documents that will be declared as records, you must create record classes in the FPOS to correspond with those document classes. There is no requirement to have a one-to-one mapping between records-enabled document classes in the ROS and record classes in the FPOS. In some solutions, a one-to-one mapping might be easier to implement. However, it can be more efficient to map the divergent metadata from the source document classes into a single or reduced set of record classes that store only the essential common properties for managing retention. 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 5-5 on page 132 shows *z* customer-defined record classes in the FPOS (the record classes starting with ILG). This record class is defined in the RB FPOS object store, which is the FPOS for our case study. This corresponds to the customer-defined document class in the ROS that we show in Figure 5-4 on page 130.

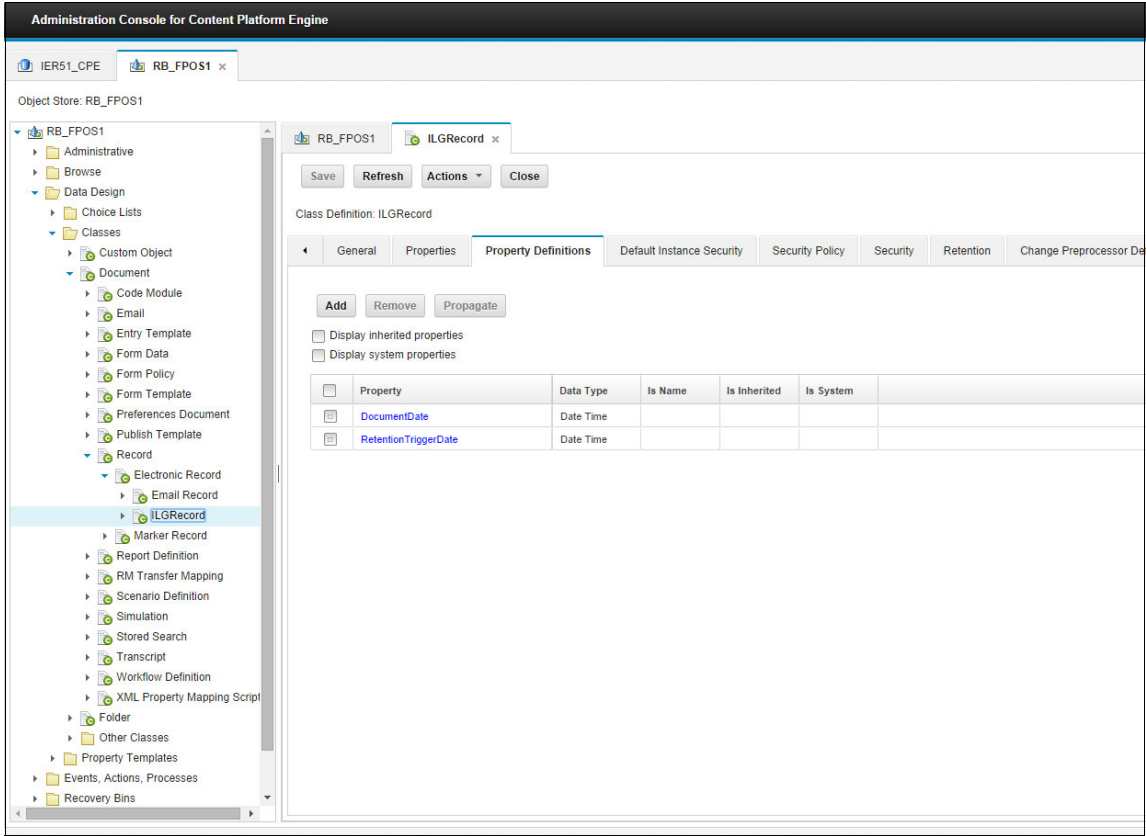


Figure 5-5 A customer-defined record class

When you define a record class, you also need to determine the properties associated with the class. The choice of properties is primarily guided by management and identification requirements for record dispositions and holds and for search and retrieval in Enterprise Records.

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

5.2.2 Manual declaration

In most organizations, users working or creating records to be manually added to IBM Enterprise Records are knowledge workers. The primary role of these users might vary vastly from that of the organizations Records Administrators and Records Managers, so for the records management system to provide a complete picture of the organizations records, it must be simple enough to use that it is embraced by all users, which is the challenge.

Users creating documents can typically use keyboard shortcuts, for example CTRL+S, to save documents, and they are prompted for the name of the document. To address the record metadata requirements, and from an IBM Enterprise Records perspective, there are now additional fields to be completed when manually adding documents, specifically the document class metadata, and the record class metadata. Both of which add more keystrokes and mouse clicks.

To overcome these challenges, IBM Enterprise Records can use many of the powerful capabilities of the underlying Content Platform Engine to assist users with manual record creation and capture, for example:

- ▶ Document entry templates
- ▶ Inheritance
- ▶ Workflow
- ▶ Document lifecycles
- ▶ Record entry templates

Each of these are described in the section that follows.

5.3 Manual record creation and capture

IBM Enterprise Records supports the ability for users to add new documents to the repository, and store the item as a record on capture into the system, or at a point in time, when the document is finalized. This can be achieved through several mechanisms including:

- ▶ Entry templates
- ▶ Events and subscriptions
- ▶ Workflows

Entry Templates provide a simplified mechanism for users to register new items into the repository, by removing the requirement for the user to specify:

- ▶ Where the item will be stored
- ▶ The class of document
- ▶ Providing default values for metadata
- ▶ Applying security to the item
- ▶ (Optional) Launching any relevant workflow for the document

Entry Templates are items secured and stored in the Content Platform Engine, so it is possible to create different templates for different groups of users, and control which users have access to the different entry templates.

Preferred practice is to use Entry Templates when there is a requirement for manual declaration of records.

The Content Platform Engine events provide a mechanism for initiating actions that are invoked when objects are created, modified, and deleted from, an object store. For example, creating a document in an object store triggers a create event, which can launch a workflow that approves the new document and declares the approved content as a record.

A subscription is the association of a particular event trigger with an event action. In the previous example, the event trigger is the document creation and the event action is the workflow launch. Many different subscriptions can be associated with a particular event trigger.

Within a Content Platform Engine workflow, the workflow can declare an attached document as a record, filed into the FPOS.

5.3.1 Document entry templates

A document entry template is an entry template which supports simplifying documents registered into the ROS. Document Entry templates have some features which are specific to documents stored in the ROS. Figure 5-6 on page 135 shows an example document entry template.

The screenshot displays a web-based form for document entry. It is divided into two main sections: 'General' and 'Properties'.
 In the 'General' section:
 - 'Entry template:' is a dropdown menu currently showing 'Add a banking document'.
 - '* Save in:' is a dropdown menu showing 'Banking'.
 - 'File name:' is a text field containing 'Deposit Slip.pdf'.
 In the 'Properties' section:
 - '* Class:' is a text field containing 'ILG Document'.
 - 'Document Title' is a text field containing 'Bank Deposit Slip'.
 - '* Account Number' is a text field containing 'ddd-ddddd-ddddd'.
 - '* Account Name' is an empty text field.
 - 'DocumentDate' is a date field containing '10/10/2014' and a time field containing '00:00'.
 Red asterisks (*) next to 'Save in:', 'Account Number', and 'Account Name' indicate required fields. Question mark icons (?) are present next to 'Document Title', 'Account Number', 'Account Name', and 'DocumentDate'.

Figure 5-6 Sample document entry template

In this sample figure, you can see the user is shown information about the document that they are adding, including the document title, and is prompted for the account number and account name, and for the document date. You might also notice:

- ▶ There are some red asterisks next to account number and account date, which indicate that you must supply values for this field.
- ▶ The format of the expected value for the Account Number field is displayed to the user, which might also be validated to ensure it is of the correct format.
- ▶ Because document entry templates in IBM Content Navigator also support external data services, when the user enters the Account Number, a lookup could be done to populate the Account Name field.
- ▶ The document date field allows the user to select a value for the date of the document, should it differ from the date it is added to the repository.

Support for Entry Templates is a new feature in the latest release of IBM Content Navigator.

Note: The current release of IBM Content Navigator does not support linking document entry templates with record entry templates to support a single interface for record capture and creation.

5.3.2 Record entry templates

Record entry templates are similar to document entry templates, but support the capture or declaration of a document stored in the ROS as a record in the FPOS, with the intent of simplifying the capture process for the user.

Record entry templates can be configured to minimize the amount of information a user needs to complete to declare the item as a record. Figure 5-7 on page 137 illustrates the interface for creating or editing record entry templates. When creating the entry template, you identify the following:

- ▶ The name of the entry template and who has access to use the template
- ▶ The location in the file plan where the record will be declared into, which can be obscured from the user, or display it as read-only
- ▶ The metadata the user needs to set, if any, and the label of the fields, for those fields the user is shown

As with document entry templates, metadata can be validated in the template using external data services to confirm completeness and correctness of the data at the point of entry into the system.

Record entry templates are created within the IBM Content Navigator user interface, by authorized users.

Record entry templates accessed by users in through the IBM Content Navigator interface.

IBM Enterprise Records

Entry Template Manager *New Record Entry Template x

Save and Close Copy Cancel

Record Entry Template Settings

Set File Plan Location

Set Record Properties

Record Entry Template Settings

* Name: ? Declare 2014 Banking Activities Record * Save in: RB Content Repository 1

Description: ? Use to declare a new banking activities record for 2015 Inherit the security settings ?

Share with: Only me Select ...

Apply the record entry template: ?

☒ Record properties dialog ?

Set File Plan Location

Specify where records are declared to.

☐ Show file plan location ☐ Hide file plan location ☒ Make file plan location read-only

* File plan repository: ? RB Record Catalog 1

Default location: ? 2014 Banking Activities Clear

Starting location: ? Clear

Set Record Properties

☒ Show the Properties section ? ☐ Hide the Properties section ?

Class: ILGRecord Edit Layout... Clear

Document Title: ?

Description: ?

DocumentDate: ? dd/MM/y HH:mm

RetentionTriggerDate: ? dd/MM/y HH:mm

12/01/2015 22:03 - 2 items were found

Figure 5-7 Sample record entry template

5.3.3 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 Enterprise Records, the record must be classified (cataloged, filed, and categorized are also common terms for classified) into a file plan. Chapter 3, “Retention and file plans” on page 65 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.

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 demonstration system showcases three examples:

- ▶ *Government Entity Financial Services Banking Activities records:* are organized by calendar year and use record folders to aggregate all records with the same calendar year to aggregate all records with the same year into one record folder for each calendar year. When a new banking activities record is added, the system must know both the category and the specific record folder in which to file the record.
- ▶ *Government Entity Equipment Insurance Renewal records:* are filed directly into the appropriate record category based on being an equipment insurance renewal. When a new renewal document is added, the system must know in which category to file the record. Renewal records do not use any record folders to aggregate records. All records are filed directly into a category, and we rely on the Policy ID property to identify individual records.
- ▶ *Government Entity Financial Services 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 5-8 on page 139, Banking Activity records are filed into a record folder. Notice that under the Banking Activities category, there are three record folders, one for each calendar year. When a Banking Activity document is declared, the calendar year determines which of the record folders is used.

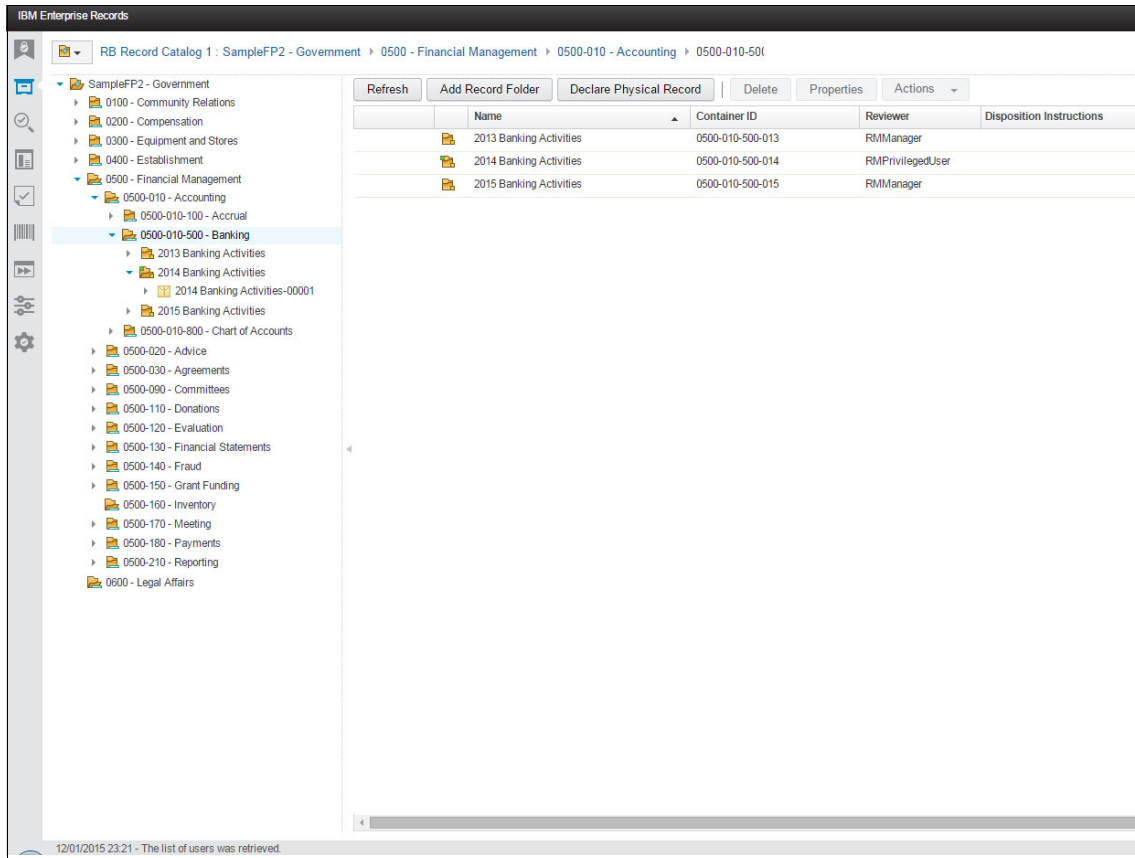


Figure 5-8 Banking Activities records are filed into a Banking Activities record folder

File plan path

One of the easiest ways for the IBM Enterprise Records 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 Equipment and Stores - Refrigerator, this is the file plan path:

/Records Management/SampleFP2 - Government/0300 - Equipment and Stores/0300-130 - Installation/0300-130-090 - Equipment - Refrigerator

Figure 5-9 shows the refrigerator folder under Equipment and Stores.

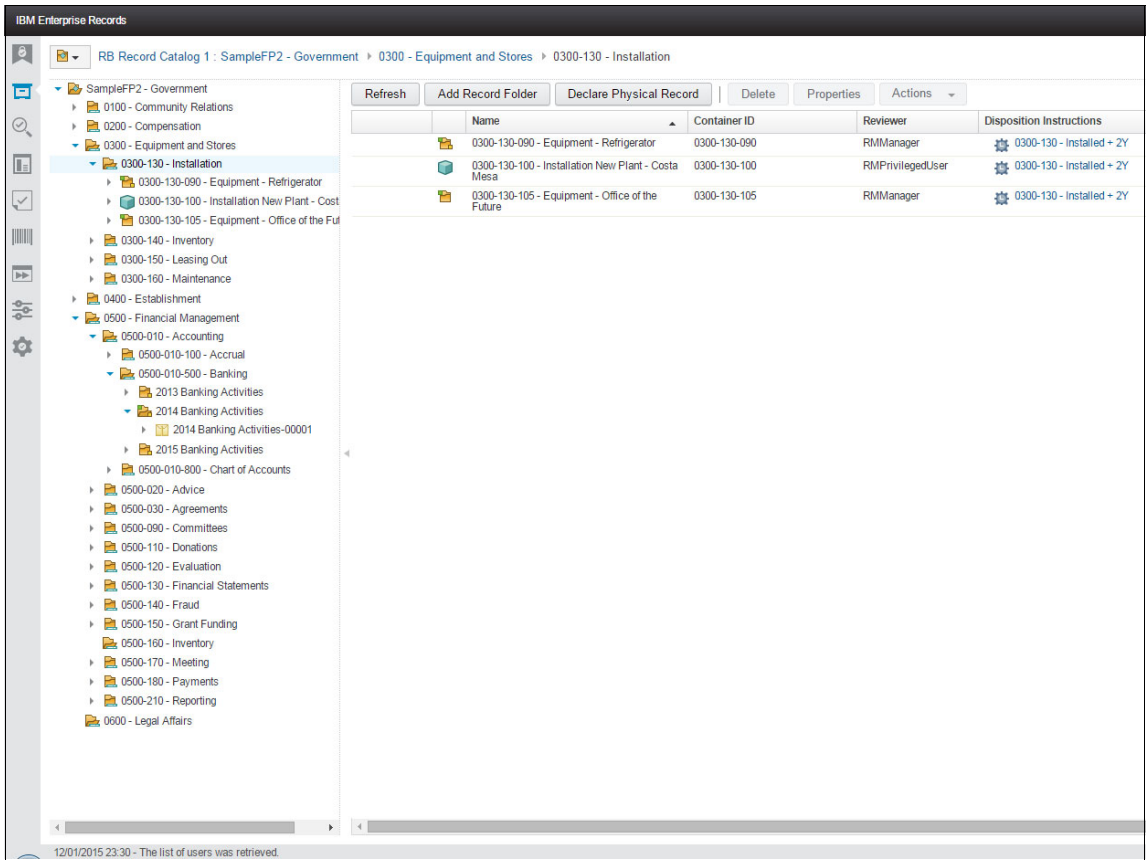


Figure 5-9 Record folders are used to aggregate equipment, such as the Refrigeration folder

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 5-10 on page 141 shows the properties that a user must enter when creating a new document of the ILGDocument class.

General

Entry template:

Add a new installation document

What do you want to save?

Local document

File name:

Choose Files

WMAN_FDFRI_IW_Mar14.pdf

☒ Major version

Properties

Class: ILG Document

Document Title

WMAN_FDFRI_IW_Mar14.pdf

Document Date

07/10/2014 00:00

Asset ID

123456

Asset Name

Refrigerator

LocalScheduleID

Figure 5-10 Document properties can often be useful for determining record classification

Continuing with our equipment example, you can see (from Figure 5-10) that Asset Name can be used to determine the file plan path for record declaration. Figure 5-11 shows how these properties can be used to construct the file plan path for record classification.

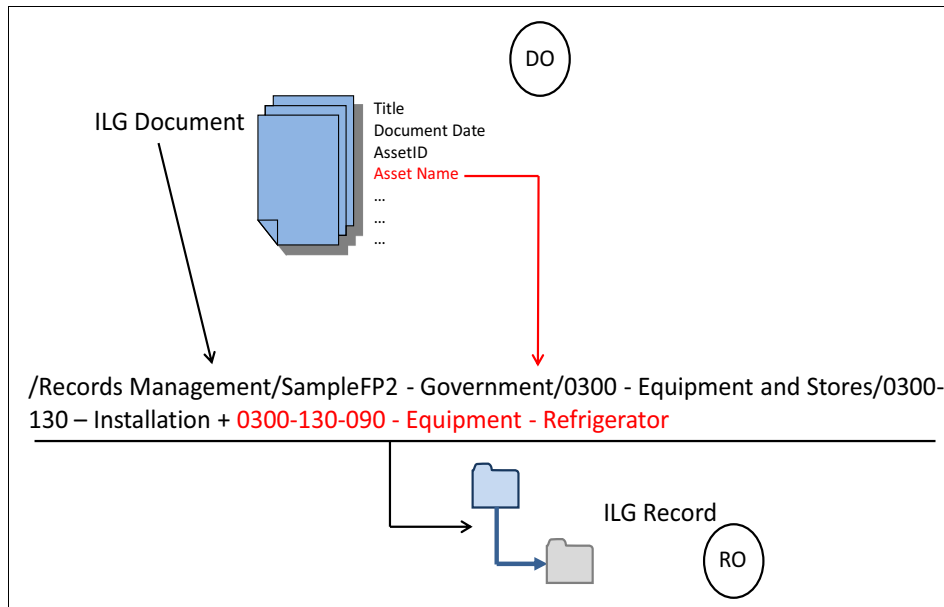


Figure 5-11 Document properties can determine the record classification

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.

5.3.4 Primary mechanism for manual ingestion and declaration

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

- ▶ IBM Content Navigator
- ▶ Entry Templates
- ▶ Using workflow

These mechanisms all work with core IBM FileNet Content Platform functions.

IBM Content Navigator

IBM Content Navigator is the primary user interface for users of the IBM FileNet Content Platform. Within IBM Content Navigator users can:

- ▶ Add documents
- ▶ View documents
- ▶ Update documents
- ▶ Update document metadata
- ▶ Work with documents
- ▶ Declare documents as records
- ▶ Search for documents
- ▶ Work with document workflows
- ▶ Perform other actions as authorized

The add document option is available to users in IBM Content Navigator according to the configuration of the desktop, but generally is available as shown in Figure 5-12 on page 143, from either the “Add Document” button or from the options menu.

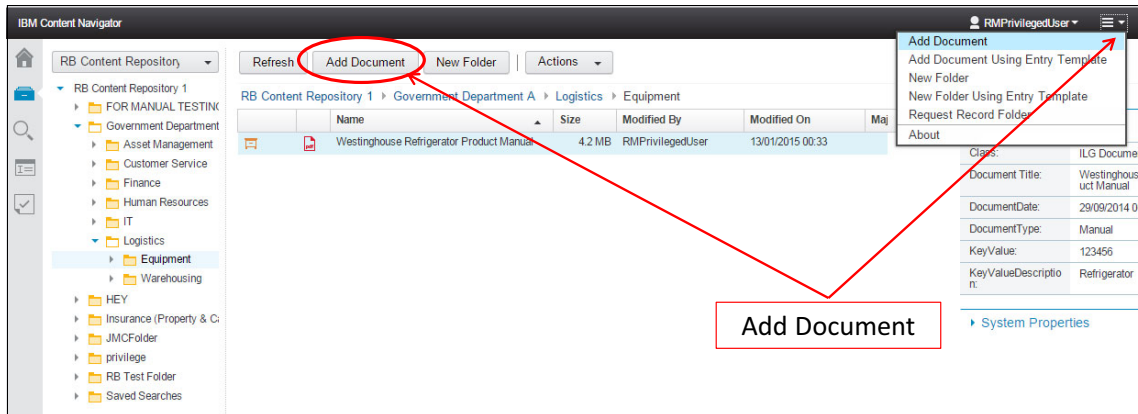


Figure 5-12 Using IBM Content Navigator to add a document

When a user adds a document through IBM Content Navigator without using an entry template, the user is guided through a document entry wizard, as shown in Figure 5-13 on page 144, which requires the user to:

1. Identify where the document will be filed.
2. Identify what will be filed, for example a document or a reference.
3. Identify if the document will be added as a major version.
4. Identify the document class, in our example ILG Document.
5. Complete the document metadata.
6. Decide if the document should be declared as a record.

Where a user decides it is relevant to declare the document as a record, they might do so manually by selecting the declare option for the document from the documents menu, or with the document highlighted and selecting the declare option from the “actions” button.

General

* Save in: Equipment

What do you want to save? Local document

* File name: Choose Files No file chosen

☒ Major version ?

Properties

* Class: ILG Document

Document Title: ?

DocumentDate: ? dd/MM/yy HH:mm

DocumentType: ?

KeyValue: ?

KeyValueDescription: ?

RecordSeriesCode: ?

RetentionTriggerDate: ? dd/MM/yy HH:mm

OrganizationUnit: ?

LocalScheduleID: ?

Figure 5-13 IBM Content Navigator Document Entry Wizard

As IBM Content Navigator includes integration to Microsoft Office, when the Navigator integration to Microsoft Office is installed, users can directly add documents to the IBM FileNet Content Platform through Microsoft Office.

Within the IBM Content Navigator user interface, users can search for and work with documents. Users are visually advised in documents have been declared as a record by cabinet icon in the far left column of the Content Navigator list view. The records management information about a document can be accessed by users leveraging the menu on declared items, or using the “actions” button when it is available in the interface, although these record properties are not necessarily meaningful to general users.

Entry templates

Document Entry templates, as described in 5.3.1, “Document entry templates” on page 134, can be used to simplify and speed up the document registration process for users, minimizing the options available to users for adding documents and reducing the number of mouse clicks and keystrokes used to capture a document.

In addition to document entry templates, IBM Content Navigator also supports the use of Record Entry templates, as described in 5.3.2, “Record entry templates” on page 136, to simplify and enhance the record declaration process for users.

When used together, and tailored for different use cases, Document Entry and Record Entry templates can greatly reduce the effort and improve consistency of record declaration for manual record creation and capture.

Using a 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 workflow using the IBM FileNet Workflow Designer tool. Such workflows can take full advantage of both Content Platform Engine operations and IBM Enterprise Records operations using the component integration mechanism. Custom operations can also be integrated if required.

Additionally, business process workflow can be combined with just about any document ingestion mechanism by leveraging the Content Platform Engine workflow subscription mechanism. No matter how a document is added to the system - whether through an entry template, IBM Datacap, IBM Content Collector tools, or through IBM FileNet Content Federation Services for 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 workflows 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 workflow
- ▶ 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 workflow where documents go through a customer-defined workflow 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 Content Platform Engine workflow subscription feature. A workflow subscription can be configured to trigger from a variety of content events, the most common and useful events being either document check-in 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 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.

Within the FileNet Content Platform, you can use workflow tool to automate the declaration of documents that are associated with a workflow as a record in IBM Enterprise Records.

The declaration process can be largely automated using the workflow tooling through use of workflow component steps which can be used to identify the file path for the location in the file plan where the record is to be stored, and ensuring relevant information is captured on declaration.

Record declaration in a workflow uses a workflow operation available in the declare record operation available in the component step for record management operations. There are certain workflow properties that must be available to support the “declare record” workflow operation, which includes:

- ▶ The record folder location
- ▶ The document to be declared as a record
- ▶ A record type indicator to advise as to what type of record is to be declared, for example, an electronic record or a marker record

- ▶ An array containing the names of the record properties which will be set
- ▶ An array containing the values of the record properties to be set

IBM Enterprise Records includes a sample declare record workflow which might be used to support workflow declaration of records.

Declaring an item as a record in a workflow is an example of how an automated process might be used to support manual record declaration, in that, the user might only identify a document which they are working on, and the declaration might occur automatically as a part of the workflow.

5.3.5 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 Enterprise Records works in regard to multiple versions of a document. As with many aspects of IBM Enterprise Records, the business requirements determine how you approach documents with versions.

Declaring IBM FileNet document versions

The Content Platform Engine offers the ability to declare a version of a document as a record, or to declare all versions of a document as a record. However, IBM Content Navigator offers a single declare feature for users, which supports the ability to declare a single version of a document, declaring the current version of the document as a record. To declare all versions of a document as a record, either a custom action must be made available in IBM Content Navigator, or an automated record declaration function must be used.

While it is possible to declare records one or more times throughout the life of that document, it is important to understand that each time that a record is declared, whether it references a single version or multiple versions, these versions are locked down by IBM Enterprise Records. Any other versions, major or minor, that have not explicitly been declared, are not locked down and are not under IBM Enterprise Records 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 superseded, which is a useful function when dealing with versioned documents in IBM Enterprise Records.

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 documents where you do not expect users to perform a check-out and check-in 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 email 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 email 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 email 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 Content Platform 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 Enterprise Records 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 must 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 Enterprise Records 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 supersede the record of the previously declared version. However, the system does not do this step automatically. A user can manually supersede an existing record, or you can write custom code to automate this requirement.

5.4 Performance considerations

The speed at which the system can ingest and declare records is an important consideration when planning the design and size of your system. When sizing your system, consider both average and peak ingestion rates, and 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 IBM Content Federation, or IBM Content Collector, that can potentially generate a continuous stream of large volumes of data. To achieve the best performance for bulk declaration scenarios, use the BDS rather than the IBM FileNet Records Manager API to declare records from a custom application.

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.



Records disposition and basic schedules

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 describe basic disposition schedules. Advanced disposition schedules are covered in the next chapter.

In this chapter, we cover the following topics:

- ▶ Introduction to records disposition
- ▶ Implementing records disposition policies
- ▶ Basic disposition schedules

6.1 Introduction to records disposition

As already described in Chapter 3, “Retention and file plans” on page 65, retention rules and policies describe how long records must be retained before disposition. Retention periods are normally defined and derived from the laws, policies, and regulations that apply to the business or organization, by jurisdiction and the geographical regions in which they operate. Additional internal policies and controls can augment these retention periods. *Disposition* refers to the actions taken on records that have reached the end of their retention periods or schedules.

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 proper disposition of records.

The type of record also plays a key role in determining what to do with a record. Record disposition is about timely destruction, but it can also be about the preservation and archiving of certain records. For example, a US government agency might be required to transfer historical records to the National Archives 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 or action, 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 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 usefulness, was not viewed by companies as a high priority.

Today, this approach (or lack of approach) to managing records is no longer acceptable. It could easily lead to enormous amounts of information being stored too long, when, instead, the information could and should be disposed of according to the applicable rules. Not only is the amount of information that organizations must manage growing exponentially, but, more importantly, the potential cost and liability associated with eDiscovery and regulatory review also rises.

An organization that either does not have or does not enforce its records management policies is exposing itself to significant liabilities:

- ▶ Spoliation charges
- ▶ Discovery liabilities
- ▶ Discovery costs
- ▶ Fines and penalties

Savvy legal counsel for plaintiffs knowingly use high discovery costs as a settlement tactic. They know that the cost and resources associated with searching through a mountain of information are prohibitive. It does not matter if there is nothing relevant in that mountain; you still must look through it. Litigation lawyers also know that, if given the chance to look at everything (unrestricted access to all of the other party's records), they have a high probability of finding damaging information that will help their cases.

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	<ul style="list-style-type: none"> ▶ Exposure to liability ▶ Target for discovery 	<ul style="list-style-type: none"> ▶ Costly to discover ▶ High storage costs
Destroying records too soon	<ul style="list-style-type: none"> ▶ Potential for spoliation charges ▶ Inability to produce 	<ul style="list-style-type: none"> ▶ Potential fines ▶ Legal costs
Lost records	<ul style="list-style-type: none"> ▶ Potential for spoliation charges ▶ Charges of ineffective records management ▶ Inability to produce 	<ul style="list-style-type: none"> ▶ Potential fines ▶ Legal costs

If there is not a clear legal, regulatory, or business justification for retaining information, you must dispose of the information.

6.2 Implementing records disposition policies

IBM Enterprise Records offers two primary options for implementing the records disposal process: basic schedules and advanced schedules:

- ▶ *Basic schedules* are intended for use cases where the disposition process requires only the simple destruction of electronic records.
- ▶ *Advanced schedules* offer the full set of features and options available for a variety of use cases, including review of vital records, destruction of physical records, record export and transfer, or any disposition process that requires multiple phases or advanced features.

Whether your solution requires basic or advanced disposition, you must schedule *sweeps* to start the disposition process and, depending on the sweep options that you select, you might be required to add workflow steps to complete the records disposition process. In the remainder of this section, we briefly describe the difference between the basic and advanced disposition schedules and introduce the concepts of *disposition sweeps* and *disposition workflow processing*. More detail follows in subsequent sections.

6.2.1 Basic disposition

Basic disposition schedules were introduced into IBM Enterprise Records to allow for more efficient disposition processing where large volumes of records must be processed efficiently without creating a performance drain on the system and where the disposition process was limited to the destruction of electronic records.

A basic schedule is simply a record category that specifies the retention period and the retention trigger property that will be applied to all records contained within that category. Think of the basic schedule as a simple container for all the records that must be grouped together under a single retention rule and processed together for disposition.

The characteristics of basic disposition schedules include:

- ▶ A more efficient sweep process to identify and destroy individual records
- ▶ A simple and direct destruction process for electronic records
- ▶ Support for record level aggregation only
- ▶ A process that supports more efficient processing for high volumes

Basic schedules work well when integrating IBM Enterprise Records with a retention policy management tool such as IBM Global Retention Policy and Schedule Management. With such integration, the retention policy layer is completely defined and maintained outside of IBM Enterprise Records and

schedules are syndicated automatically to IBM Enterprise Records to implement the policy in a given records repository. Basic schedules can also be useful, and might even be the preferred choice, for a variety of common use cases even without such integration.

6.2.2 Advanced disposition

Advanced disposition schedules are the full-featured disposition schedules that have been part of the IBM Enterprise Records software since its inception. These disposition schedules are defined separately from the record categories to which they are applied and support a more flexible configuration.

The characteristics of advanced disposition schedules include:

- ▶ Support for all aggregation levels
- ▶ Support for schedule inheritance and propagation
- ▶ Support for multiple disposition phases and alternate retention periods
- ▶ Support for multi-filing of records
- ▶ Support for physical records
- ▶ Configuration of separate triggers and actions
- ▶ A more complex sweep process

Advanced schedules are appropriate for file plan configurations requiring any of the advanced features not supported by basic schedules.

6.2.3 Scheduling and monitoring disposition sweeps

With IBM Enterprise Records, the Tasks view is used for scheduling and monitoring disposition sweeps for both basic disposition and advanced disposition. The Tasks view is also used for scheduling reports and running hold sweeps. You can filter what is displayed in the Tasks view to show the tasks you are interested in listing.

Figure 6-1 on page 156 shows the Tasks view where you can schedule various sweeps and monitor the progress and results of disposition sweeps.

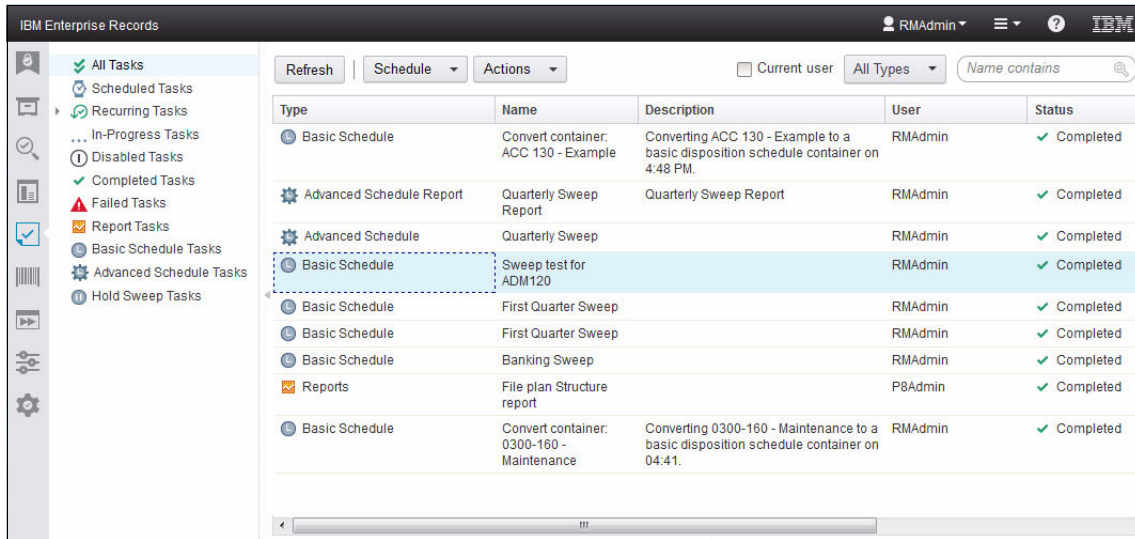


Figure 6-1 IBM Enterprise Records desktop Tasks view is used to schedule and monitor sweeps

6.2.4 Completing the disposition process

The IBM Enterprise Records desktop provides the Work view for accessing the workflow steps involved in completing the disposition process. Depending on the options that you select for basic or advanced disposition schedules, you might be required to participate in a review or approval process. These optional process steps can be accessed using the Work view and the public inboxes (workflow queues) that are set up to manage both the basic and advanced disposition workflow processes.

Figure 6-2 on page 157 shows the IBM Enterprise Records desktop Work view that was configured to display the public inboxes for records management, which include the Basic Schedule Workflow Reviewer queue and the Records Manager Approval queue.

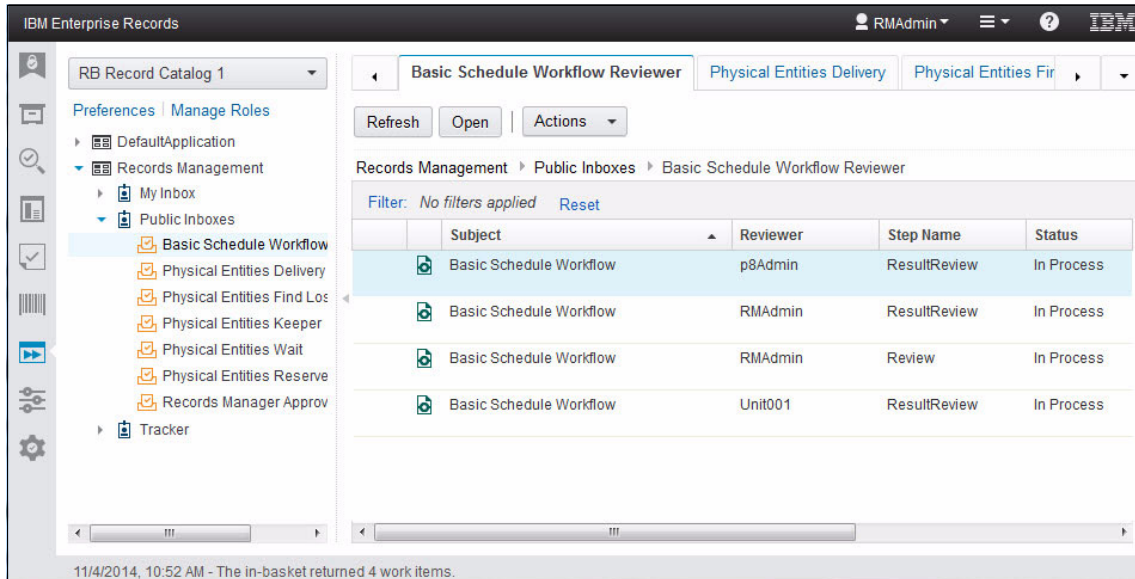


Figure 6-2 Disposition process is managed in the IBM Enterprise Records desktop Work view

In the sections that follow, we describe basic disposition schedules and processes in more detail.

Note: Advanced disposition schedules are described in Chapter 7, “Advanced disposition” on page 175.

6.3 Basic disposition schedules

In this section, we describe how to use basic disposition schedules and how the basic disposition process works.

6.3.1 Characteristics of basic disposition schedules

A basic disposition schedule is simply a record category that directly controls the disposition of the records that it contains, based on the specified retention trigger property and the specified retention period. Basic disposition schedules can contain only electronic records, and the records must each have the specified retention trigger property. Disposition processing occurs by scheduling a basic disposition sweep, which directly applies the retention properties to all of the records in the record category to determine which records are ready for

disposition. The output of the basic disposition sweep is a simple destruction eligibility report that lists each record that is ready for destruction, based on the specified retention period, as it applies to the specified retention trigger property value of each record. This report is used by the disposition process to automatically destroy each of the eligible records after an optional review period and after an optional approval step.

By its nature, a basic schedule is a leaf node record category in the file plan hierarchy. The parent record category of a basic schedule has no schedule defined. You cannot add any record categories or record folders to a record category that has been configured as a basic schedule. A file plan can contain both basic disposition schedules and record categories with advanced schedules. Therefore, to avoid confusion, it is helpful to organize the file plan hierarchy in a consistent manner.

Figure 6-3 shows one example from our financial institution use case where the record categories for each record series under Banking are configured as basic schedules. The retention trigger property name for each of the categories is shown under the Disposition Instructions column in the list view panel. In this example, all records contained in the BNK100 record category will be processed together under the same basic disposition schedule. This way of organizing basic schedules is sometimes called the “big bucket” approach, where all records that belong to the same record series are placed in the same container.

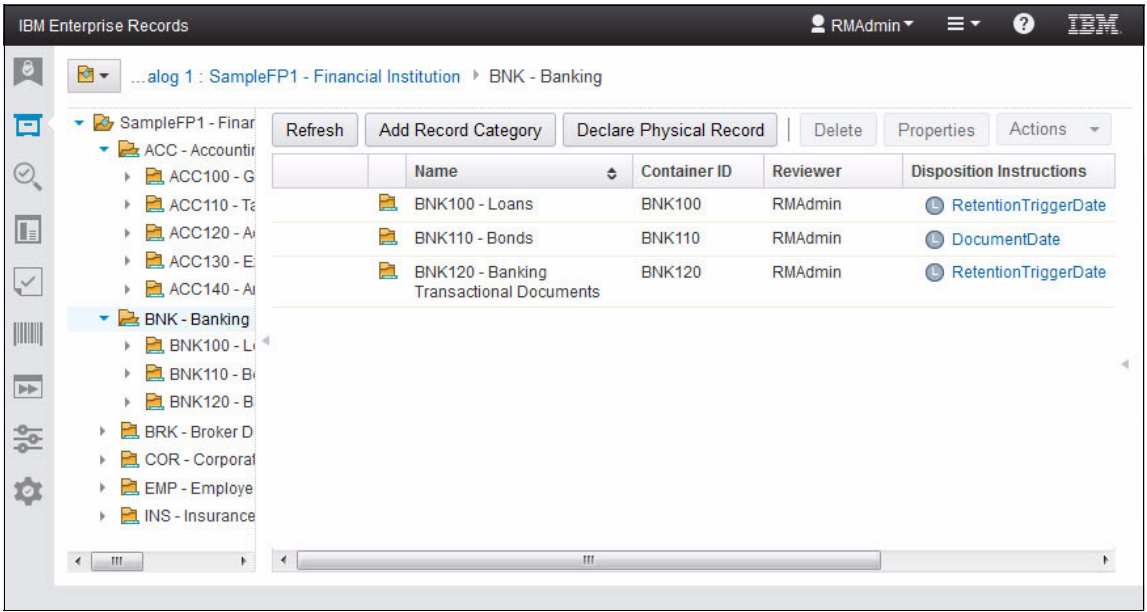


Figure 6-3 Record categories configured as basic disposition schedules

Another important aspect of basic disposition schedules is the use of the Reviewer property on the record category to group records for processing. The basic disposition sweep uses the Reviewer property from the record category that contains the records rather than using the Reviewer property from each record to group the records eligible for the destruction process. This means that all eligible records contained in the same record category will be processed together and cannot be separated for review or approval if those steps are required.

Figure 6-4 shows an example in which basic schedules are organized by business unit. Separate record categories are established for each business unit for the same record series, in this case, ADM120. By setting a value for the Reviewer property that identifies each business unit, the disposition sweep separates the records based on this property so that the records for each business unit can be reviewed and approved for destruction separately.

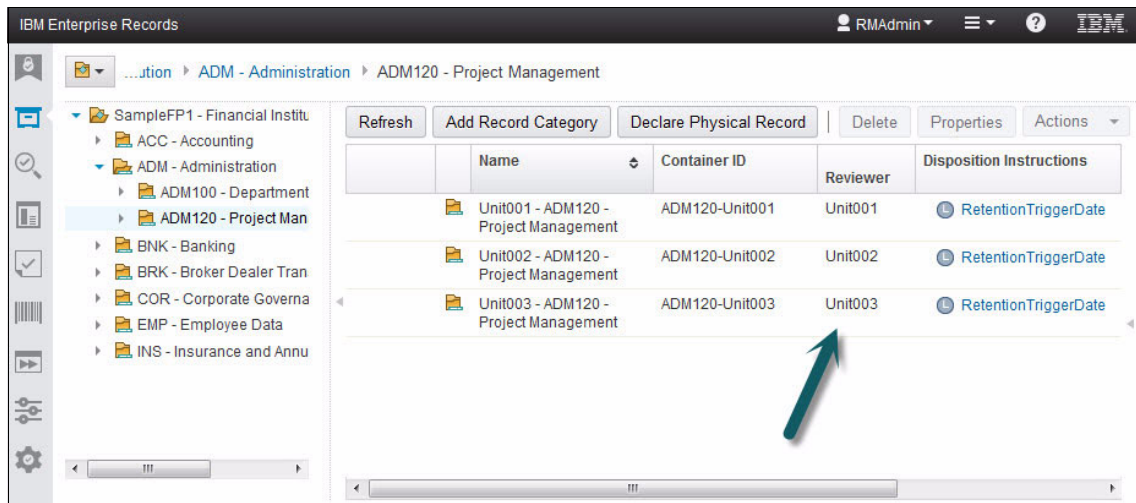


Figure 6-4 The Reviewer property for each basic schedule determines how records are grouped for disposition

In this example, each business unit declares and maintains their own ADM120 - Project Management records. In 6.3.5, “Example use cases for basic disposition sweep” on page 167, we illustrate how running a disposition sweep on the ADM120 record category results in separate destruction eligibility reports and separate disposition processes for each basic schedule in this record series.

6.3.2 Creating a basic disposition schedule

You can create schedule by selecting the **Basic disposition schedule** option when adding a new record category to the file plan. The record category is then synonymous with the basic schedule; they are not separate objects. The basic schedule is the combination of two properties that define the retention for the records in that category.

There are two properties that define the basic schedule for a record category:

- Retention trigger property name** The symbolic name of the DateTime property that must be available on each record
- Retention period** The period of time to keep each record from the specified date

Figure 6-5 shows the selection of a basic disposition schedule when adding a new record category.

The screenshot shows a configuration window titled "Disposition". Below the title is a descriptive paragraph: "Basic disposition schedules are high-performance schedules that are easy to use. Advanced disposition schedules offer more capabilities and flexibility, but they are more complex and negatively impact performance." There are two radio buttons: "No schedule" (unselected) and "Basic disposition schedule" (selected). Below the "Basic disposition schedule" option, there are two required fields: "Retention trigger property name" with a dropdown menu showing "DocumentDate", and "Retention period" with three spinners for "Years" (set to 6), "Months" (set to 0), and "Days" (set to 0). Below these, there is an unselected radio button for "Advanced disposition schedule". Under the "Advanced disposition schedule" option, there are two text input fields: "Disposition instructions:" and "Disposition authority:". The "Basic disposition schedule" option is highlighted with a light blue background.

Figure 6-5 Defining a basic disposition schedule

A record category that has been defined with a basic schedule can then be used to contain records that have the specified retention trigger property. For the schedule to work properly, all records that are declared in that record category must have the specified retention trigger property. The schedule uses the specified property, along with the specified retention period, to determine which records contained in that category are ready for disposition.

Figure 6-6 shows the detail for the ACC120 record category that was configured as a basic schedule.

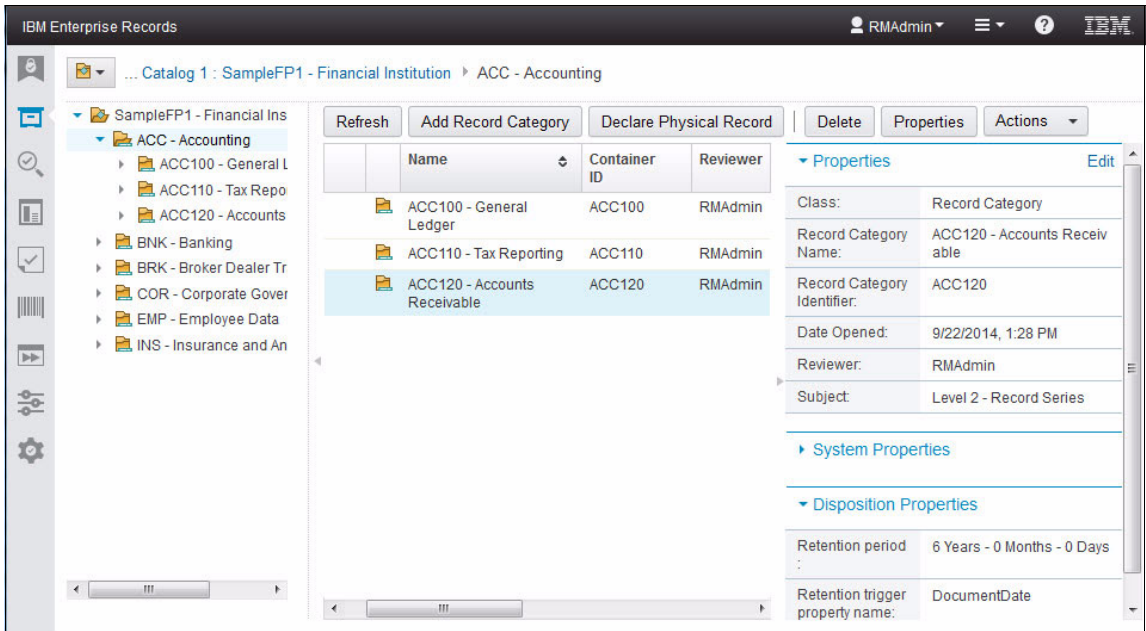


Figure 6-6 The disposition properties show the retention period and the retention trigger property name

6.3.3 Converting a record category to a basic schedule

Record categories that have either no schedule assigned or that have previously been configured with an advanced disposition schedule can be converted to a basic schedule. To convert a record category to a basic schedule, the record category must meet the following conditions:

- ▶ It must be a leaf node in the file plan tree hierarchy.
- ▶ It must contain only electronic records that are not multi-filed if it contains any records.
- ▶ All records contained in the category must have the retention trigger property that you intend to assign to the basic schedule.

To convert a record category to a basic disposition schedule, you must schedule the conversion task. After the conversion is complete, you cannot reverse the process. An advanced schedule can no longer be assigned to that record category. Depending on how many records are in the category at the time of conversion, the conversion process might take a long time, because each record must be validated to ensure that it has the appropriate retention trigger property.

There are typically two reasons for converting a record category to a basic schedule configuration:

- ▶ You have created record categories with no schedule assigned, and you are ready to assign the basic schedule configuration.
- ▶ You have created record categories that have previously been configured with advanced disposition schedules, and you want to use the more efficient and simplified basic disposition process.

Existing deployments that use record level aggregation can potentially convert their leaf node record categories to basic schedules if the existing file plan structure meets the requirements for basic disposition.

6.3.4 Basic disposition sweep and processing

To process records for disposition, a basic disposition sweep must be scheduled. To schedule the sweep, you must specify parameters for the sweep and then set the schedule for the sweep. The sweep is scheduled from the IBM Enterprise Records desktop Tasks view.

Set the sweep parameters

The following parameters are available when scheduling a basic disposition sweep:

File plan repository	You must select the file <i>plan object store</i> (FPOS) that you want to sweep.
Containers for sweep	You must select one or more containers to sweep.
Report review period	This is the number of days that the Retention Due report will be available for review during the destruction process. Zero is the default value and indicates no review period. This value is also used to postpone the disposition cutoff from the date when the sweep is run.
Report only	(Yes or No) The choice determines whether to generate only the destruction eligibility report, without actually initiating the destruction process, or to generate the report and then launch the destruction process.
Workflow connection point	This is the connection point that is used to launch the destruction workflow.

Basic schedule workflow	This determines the specific workflow to launch if more than one is available.
Needs approval	(Yes or No) The choice determines whether approval is required before the destruction occurs.

Figure 6-7 shows the parameters that are available for scheduling the basic disposition sweep.

The screenshot shows a web-based configuration interface titled "Set Parameters for Basic Disposition Schedule Sweep". It contains several parameter fields, each with a red asterisk indicating it is required. The fields are: "File plan repository" with a dropdown menu showing "RB Record Catalog 1"; "Containers for sweep" with a text box containing "SampleFP1 - Financial Institution" and a "Select" button; "Report review period" with a text box containing "0"; "Report only" with a dropdown menu showing "No"; "Workflow connection point" with a dropdown menu showing "v520Ros2_CP522"; "Basic schedule workflow" with a text box containing "Basic Schedule Workflow (Version 1.0)" and a "Select..." button; and "Need approval" with a dropdown menu showing "No". Each field has a small question mark icon to its right.

Figure 6-7 Basic disposition sweep parameters

Set the sweep run schedule

The basic disposition sweep also has the following set of options for setting the sweep run schedule:

- *Name*: You must provide a name for the sweep that will be used to help identify the sweep in the task view.
- *Description*: You can include a more detailed description explaining the nature of the sweep.

- ▶ *Schedule*: You can select whether to run once or run on a schedule.
 - *Run once*. You can specify a start time or choose to start immediately.
 - *Run on a schedule*. You must specify the frequency, a start date and time, and, optionally, an end date.
- ▶ *Login information*: If you choose anything other than *start immediately*, you must also provide login information. When starting immediately, the sweep will use the current session credentials.
- ▶ *Notification*: (Optional) You may provide an email address for notification.

Figure 6-8 shows the sweep run schedule options for the basic disposition sweep. During initial testing and validation in a development or test environment, it is typical to schedule the sweep to start immediately. However, in a production environment, you might want to schedule the sweep to start at a specific time or on a set schedule.

▶ Set Parameters for Basic Disposition Schedule Sweep

▼ Set Schedule

▼ Schedule Information

* Name:

Test sweep on Sample FP1

Description:

Run immediately on Sample FP1 with report only|

☒ Run once

* Start time:

☒ Start immediately

☐ Run on a schedule

* Repeats:

Daily ▼

* Start date:

End date:

▼ Login Information

* User name

Figure 6-8 Basic disposition sweep scheduling options

It is a good practice to provide a meaningful name and description for each sweep because all sweeps that are scheduled will show in the Tasks view. Providing a meaningful name and description makes it easier to find the sweep results that you are looking for and to distinguish one sweep from another when

you run multiple sweeps. Depending on your solution requirements, you might choose to run a single repeating sweep on the entire file plan, or you might choose to run several different sweeps on different sections of the file plan at different times to distribute the load.

View the sweep results

When the sweep is finished, you can view the results in the **Tasks** view by selecting the sweep and viewing the **Results** tab. For a basic disposition sweep, the result will depend on whether you set the *Report only* parameter to **Yes** or **No**. In either case, the results include a Retention Due report that is stored in the FPOS repository as a transcript and can be downloaded or viewed through a link.

- ▶ If *Report only* is set to **Yes**, the process is complete, and the Retention Due report can be used for reporting.
- ▶ If *Report only* is set to **No**, the basic schedule workflow will be launched with the Retention Due report as an attachment that will be used in the destruction process for review or approval and to destroy the eligible records.

Figure 6-9 shows the results for the selected basic schedule sweep: Sweep test for ADM120. The sweep resulted in a single Retention Due report, and the basic schedule workflow was launched successfully. You can also view details of the sweep and the parameters that were selected in the corresponding tabs.

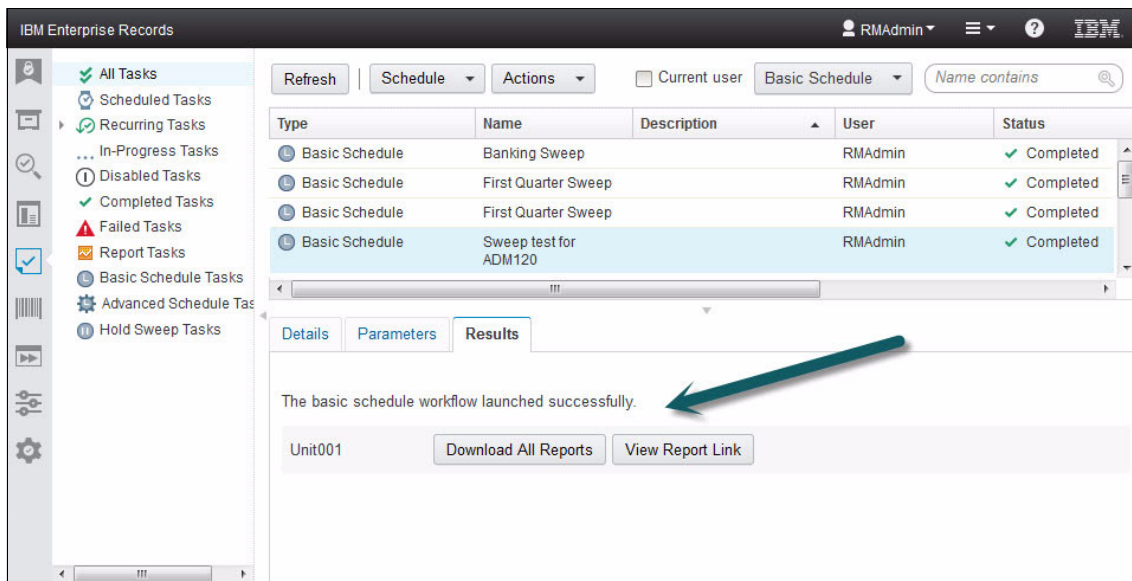


Figure 6-9 Basic schedule sweep results include a list of one or more retention due reports

Complete the disposition process

When *Report only* is set to **No**, the basic schedule workflow is launched to run the basic disposition process. This process is managed and completed through the IBM Enterprise Records desktop Work view. Depending on the options selected when setting the sweep parameters, the workflow could include optional review and approval steps before records are automatically destroyed based on the attached *retention due report*.

Review step

If a *Report review period* was specified for the sweep, the disposition process will be delayed for that number of days to allow the Retention Due report to be reviewed. The review is not required, but the delay cannot be bypassed. The work item is available in the Basic Schedule Workflow Reviewer queue. If the *Report review period* is set to 0 (zero, which is the default), the review step automatically completes as soon as the workflow is launched, without any delay.

Approval step

If *Need approval* is set to **Yes**, an approval step is required in addition to the optional review period.

Destruction step

When the review period has elapsed and the optional approval step has been completed, the basic schedule workflow automatically attempts to destroy all records identified in the Retention Due report. Any records that might have been placed on hold in the intervening time from when the Retention Due report was generated by the sweep will *not* be destroyed. The destruction step produces two Destroy Result reports that are stored as transcripts on the FPOS:

- ▶ Deleted Records report
- ▶ Not Deleted Records report

These transcripts, respectively, list the individual records from the Retention Due report that were either deleted or not deleted, depending on whether they had been placed on hold subsequently.

Result review step

When either the review period or the approval step is selected as an option, the default basic schedule workflow includes a *result review* step. This step provides access to the two transcript reports that are generated by the destruction step to allow a user to review the final results of the disposition process. If reviewing these results is not required, this step can be removed from the basic schedule workflow.

Declare record step

An option is available to automatically declare the destruction reports as records. Because the Deleted Records report and the Not Deleted Records report are stored on the FPOS, you can enable the FPOS and the transcript document class for record declaration and enable the record declaration option for the basic schedule workflow. If this option is enabled, you must specify the appropriate container for record declaration.

6.3.5 Example use cases for basic disposition sweep

In this section, we present three examples of use cases with varying parameter settings. These parameters can be combined in additional ways to achieve a variety of outcomes, depending on the requirements.

Example 1. Sweep for report only

In this example, assume that it is November 15, just six weeks before the end of the calendar year, and we want a list of records for ADM120 that will be ready for destruction at the end of the year, but we are not ready to initiate the destruction process yet. We schedule a basic disposition sweep with the following parameters and schedule:

- ▶ Containers for sweep: Select only **ADM120**.
- ▶ Report review period: Set to **45** (number of days until end of year).
- ▶ Report only: Set to **Yes**.
- ▶ Run once: Select **Start immediately**.

Figure 6-10 on page 168 shows the parameters to set for this example.

▼ Set Parameters for Basic Disposition Schedule Sweep

* File plan repository: ? RB Record Catalog 1 ▼

Containers for sweep: ? ADM120 - Project Management × Select

* Report review period: ? 45

* Report only: ? Yes ▼

* Workflow connection point: ? v520Ros2_CP522 ▼

* Basic schedule workflow: ? Basic Schedule Workflow (Version 1.0) Select...

* Need approval: ? No ▼

▶ Set Schedule

Figure 6-10 Basic disposition sweep parameters for generating a report only

This sweep simply generates a Retention Due report, without launching the basic schedule workflow. The report can be viewed independent of any disposition process, if needed.

Example 2. Sweep for destruction with a 30-day review period

In this example, assume that it is December 1, one month before the end of the calendar year, and we want to automatically destroy all records in the repository that will be eligible for destruction on January 1. However, because this sweep will run through the entire repository, we want to schedule the sweep to start after midnight when the system has a light load. We schedule a basic disposition sweep with the following parameters and schedule:

- ▶ Containers for sweep: Leave blank to sweep the entire repository.
- ▶ Report review period: Set to **30** (number of days until end of year).
- ▶ Report only: Set to **No**.
- ▶ Need approval: Set to **No**.
- ▶ Run once: Set to Dec 2 at 1:00 AM.

Figure 6-11 shows the parameters to select for this use case.

▼ Set Parameters for Basic Disposition Schedule Sweep

* File plan repository: ? RB Record Catalog 1 ▼

Containers for sweep: ? Select

* Report review period: ? 30

* Report only: ? No

* Workflow connection point: ? v520Ros2_CP522 ▼

* Basic schedule workflow: ? Basic Schedule Workflow (Version 1.0) x Select...

* Need approval: ? No

Set Schedule

Figure 6-11 Basic disposition sweep parameters for automatic destruction

The sweep runs on Dec 2 at 1:00 AM and generates a Retention Due report for all records that will be eligible for destruction 30 days from the date that the sweep was started. When the Retention Due report is generated, the basic schedule workflow will be launched and a review period will be in effect for 30 days before all of the records on the Retention Due report are automatically destroyed.

Example 3. Sweep for destruction with mandatory approval

In this example, assume that it is early December 2014, and we want to set up a recurring sweep for all Banking records (all record series under BNK in our financial institution sample file plan) before the end of the year and have that sweep run every year. We want the disposition process to have a mandatory Approval step where the records in the Retention Due report will be destroyed only if approved for destruction.

We schedule a basic disposition sweep with the following parameters and schedule:

- ▶ Containers for sweep: Select **BNK Banking**.
- ▶ Report review period: Set to **15**.
- ▶ Report only: Set to **No**.
- ▶ Need approval: Set to **Yes**.
- ▶ Run on a schedule: Set to start Dec 16 2014 at 12:00 AM.

Figure 6-12 shows the parameters to select for this use case.

▼ Set Parameters for Basic Disposition Schedule Sweep

* File plan repository: ? RB Record Catalog 1

Containers for sweep: ? BNK - Banking x Select

* Report review period: ? 15

* Report only: ? No

* Workflow connection point: ? v520Ros2_CP522

* Basic schedule workflow: ? Basic Schedule Workflow (Version 1.0) x Select...

* Need approval: ? Yes

► Set Schedule

Figure 6-12 Basic disposition sweep with mandatory approval

Because we are scheduling this sweep with a review period of 15 days and running the sweep yearly, starting on Dec 16, it will pick up any records that are eligible for destruction through the end of the year.

Figure 6-13 shows the schedule for a recurring sweep.

Set Parameters for Basic Disposition Schedule Sweep

Set Schedule

Schedule Information

* Name: Recurring annual sweep for BNK

Description: Sweep BNK once a year starting in Dec 2014 with a 15 day review period

☐ Run once

* Start time: immediately

☒ Run on a schedule

* Repeats: Yearly

* Start date: 12/16/2014 12:00 AM

End date:

Figure 6-13 Basic disposition sweep configured to run on a yearly schedule

Because we specified a 15-day review period, the records will not be destroyed until after that 15-day review period has expired *and* the records have been approved for destruction. If the records are not approved for destruction, they will not be destroyed even if the 15-day review period has passed.

6.3.6 Controlling how records are grouped for disposition

With basic disposition schedules, all of the records in a single container (record category) are grouped together in a single Retention Due report based on the Reviewer property of the record category. If the solution calls for an entirely automated destruction process with no review or approval, it is possible to use a single schedule for each record series. However, if either a review or approval is required before records can be destroyed, it will probably be important to organize the file plan with a more granular structure to separate records that belong to the same record series, based on who should complete the review or approval.

For example, as described in 6.3.1, “Characteristics of basic disposition schedules” on page 157, the file plan for our financial institution use case example shows each record series under Banking configured as basic schedules, thereby acting as “big bucket” containers for all of the records of each record series. However, in many use cases, it is desirable to separate records based on which organization unit (department, business unit, and so on) is responsible for the records. In such cases, the record series can serve as a parent category for the basic schedules that will contain the records for each organization unit separately.

In Figure 6-14 on page 173, we compare two different record series that illustrate two different ways of grouping records. In the case of the record series named BNK100 (Banking - Loans), all records are grouped together because there is no requirement to process them separately for disposition. In this example, when the retention is due for any specific loan record, it will be processed for destruction immediately without review or approval.

In contrast, the ADM120 record series must be separated into more granular record categories for each organization unit that needs to keep project management records separately. By separating the records for each unit, we control the access to the records separately and can assign a separate reviewer for each subcategory so that the records will be processed separately for review and approval.

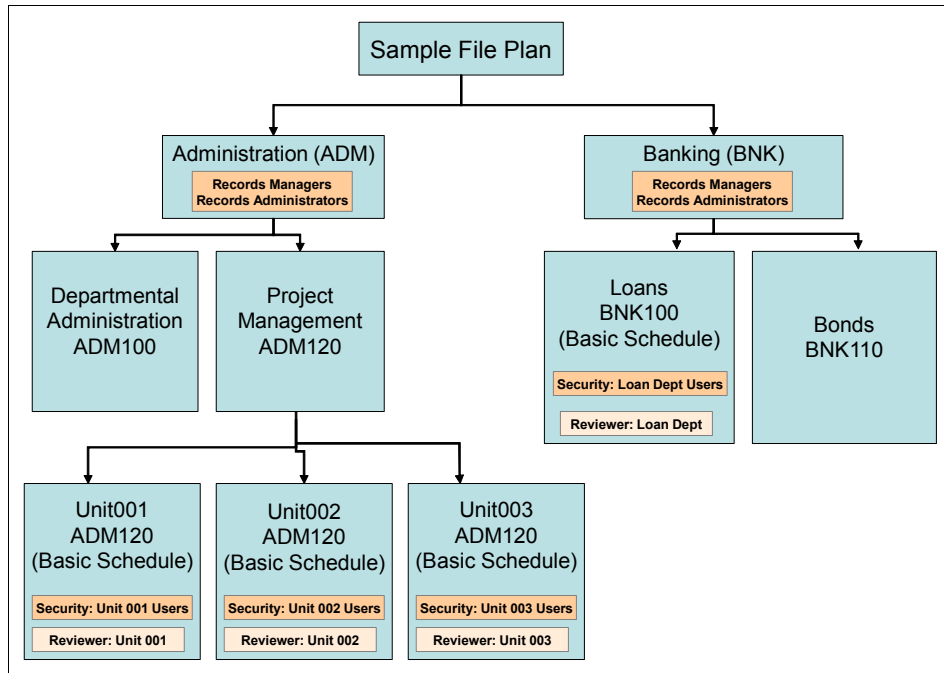


Figure 6-14 Separating record categories to group records by organization unit

We use the Reviewer property of the record category to indicate who is responsible for the records in that category. Also, the structure of this file plan aligns with the security access for the records in each category.

Figure 6-15 on page 174 illustrates this example for the ADM120 record series, where we have created a basic disposition schedule subcategory for each organization unit and assigned an appropriate value for the Reviewer property.

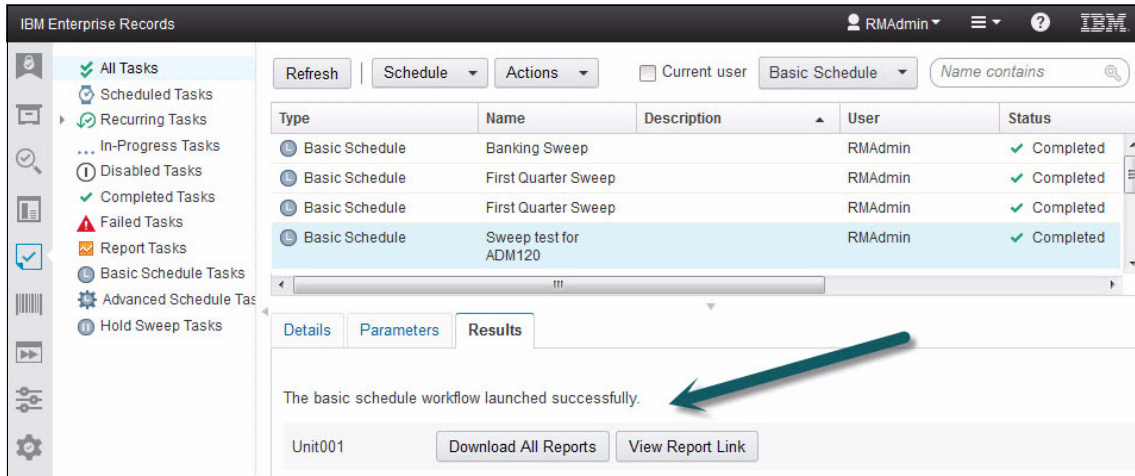


Figure 6-15 Separate record categories for each organization unit help control the disposition process

If we schedule a basic disposition sweep on the ADM120 parent category, the sweep automatically uses the Reviewer property on each subcategory to generate separate Retention Due reports. Separate disposition workflows are launched by the sweep. This enables the records in each subcategory to be reviewed and processed separately by the appropriate business units, if that is preferable.

If we schedule a basic disposition sweep on the BNK100 record category, all records with Retention Due will be processed together in a single workflow because we configured BNK100 as a basic disposition schedule. Because of the way we organized these record categories, we will get the same results even if we scheduled a single sweep for the entire file plan.



Advanced disposition

Advanced disposition includes the full set of features and functions that might be required for more complex disposition processes.

In this chapter, we cover the following topics:

- ▶ Advanced disposition schedules
- ▶ Advanced disposition sweep
- ▶ Initiating and completing disposition
- ▶ Automatic destruction using Auto Destroy
- ▶ Running a sweep from the command line
- ▶ Performance considerations
- ▶ Converting advanced schedules to basic schedules

7.1 Advanced disposition schedules

This section provides an overview of the elements and components that are related to advanced disposition in Enterprise Records and explains how these elements and components relate to the lifecycle of a record.

The following list identifies the main activities to perform to implement advanced disposition by using Enterprise Records:

1. Configure disposition schedules to represent the relevant retention rules in the organization's retention schedule.
2. Apply the disposition schedules to the appropriate categories in the file plan to affect the records that are ready for disposition.
3. Schedule a disposition sweep.
4. Initiate disposition of the categories 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. To understand the options available, we review the components of a disposition schedule in more detail.

7.1.1 Disposition schedule

Disposition schedules encapsulate the retention rules for records and instructions for the disposal of records at the end of the retention period. IBM Enterprise Records uses advanced disposition schedules to control the retention and disposal of records with a variety of advanced features and options.

The Enterprise Records advanced disposition schedule includes the following primary components:

- ▶ *Disposal trigger* that specifies a trigger condition and aggregation level
- ▶ *Cutoff* as defined by a cutoff base and cutoff delay
- ▶ One or more *disposition phases*, each of which defines a retention period
- ▶ *Actions* associated with each disposition phase

Figure 7-1 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 the cutoff date. *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 of is ready for the disposition action associated with that phase. The disposition action for a phase can then be initiated and completed.

Next, we describe each of these topics in more detail.

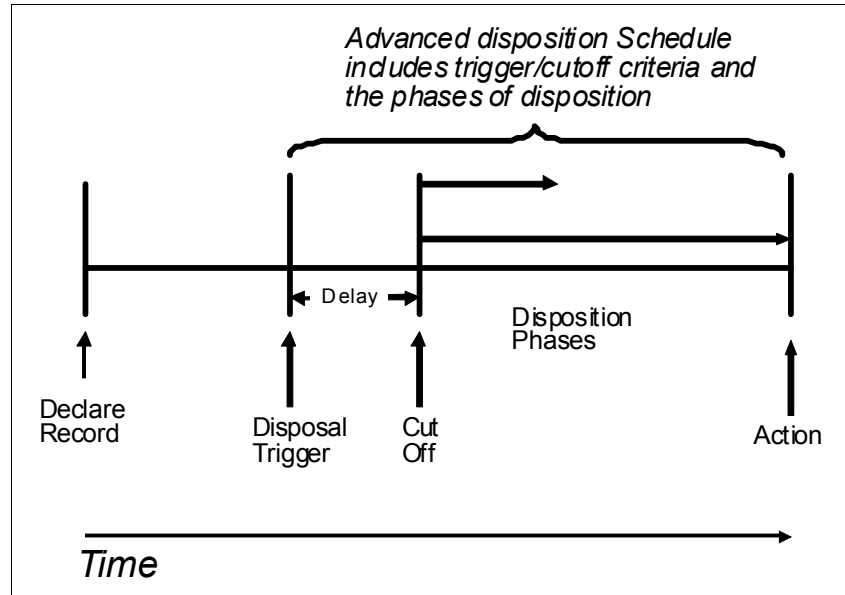


Figure 7-1 The relationship of a disposition schedule to the lifecycle of a record

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 an *advanced disposition sweep* to compute essential disposition parameters that can be calculated only after the trigger condition is met. We describe the role and use of advanced disposition sweep in a later section.

Before we describe the various configuration elements and how they come together in a disposition schedule, it is instructive to review 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 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.

Preferable 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 enterprise 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 used solely for purposes of documentation. In Enterprise Records, 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 7-2 on page 179 shows a list of the advanced disposition schedules that we have configured for our sample government file plan. We chose to name the schedules by record series code and to include concise descriptive information about the trigger and retention period in the schedule name. The disposition authority in this example is populated with a governmental regulatory agency code that serves as the citation for each schedule. A more lengthy description is provided to describe the nature of the trigger and the disposition process in more detail, if needed.

Schedule Name	Disposition Authority	Description
0300-130 - Installed + 2Y	AFDA1149	Destroy 2 years after equipment is installed and configured
0300-150 - Closed + 7Y	AFDA1155	Destroy 7Y after lease expiration
0500-010 - Received + 7Y	AFDA1219	Auto Destroy 7 years after document is received
0500-030 - Expiration + 7Y	AFDA1215	Destroy 7 years after expiration or termination
0500-130 - Received + 7Y	AFDA1250	Destroy 7 years after the document is received
0500-150 - Acquitted + 7Y	AFDA1253	Destroy 7 years after grant acquittal
0500-160 - Received + 2Y	AFDA1254	Auto Destroy 2 years after document is received
0500-170 - Received + 7Y	AFDA1239	Destroy 7 years after document is received

Figure 7-2 Advanced disposition schedules

Next, we describe the various configuration elements that define the behavior of the disposition schedule.

7.1.2 Disposal triggers

A *disposal trigger* signals that record disposition can begin. Often, the trigger condition 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 allows the remaining disposition parameters to be calculated. For example, an employee document might be triggered for disposition by the employee termination date. While an employee is still active and employed by the organization, the date value remains null. As soon as the employee termination 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 of (an individual record, a volume, a record folder, or an entire record category).

In Enterprise Records, a disposal trigger can be defined as one of three types:

- *Internal event trigger*. These are the most commonly used triggers. They are tied to the metadata of the entities being disposed of, whether that metadata is internal system metadata or trigger data that is populated from an external system. After the trigger condition is satisfied, a sweep can calculate the

remaining disposition parameters that determine cutoff and the retention period. Internal event triggers are typically based on a Date property on the entity being disposed of. The scenarios for the case studies used in this book rely on internal event triggers as do most real-world use cases.

Examples: Expiration date, received date, and closed date

- ▶ *External event trigger*: External event triggers are essentially a fixed date value that will apply to all records that the date affects. This type of event trigger is useful for only broad application of a single date value to all records affected by a disposition schedule. Even though it is called an external event trigger, this type of trigger does not lend itself to many of the typical use cases where the trigger data comes from an external system.

Example: Life of Company A

- ▶ *Recurring event trigger*: Recurring events are events that recur automatically after a predefined time interval. They are associated with vital records and are used to trigger periodic reviews of these records.

Example: Annual reviews

Note: Most use cases that involve data from an external system will use the *internal event trigger*, because the data that comes from the external system updates properties in Enterprise Records that are treated as internal. Many people refer to a data feed from an external system as the “external event triggers” or “event-based retention triggers,” even though they do not use the external event trigger feature in Enterprise Records. It is a best practice to implement these as internal event triggers because the properties are updated in Enterprise Records.

Aggregation

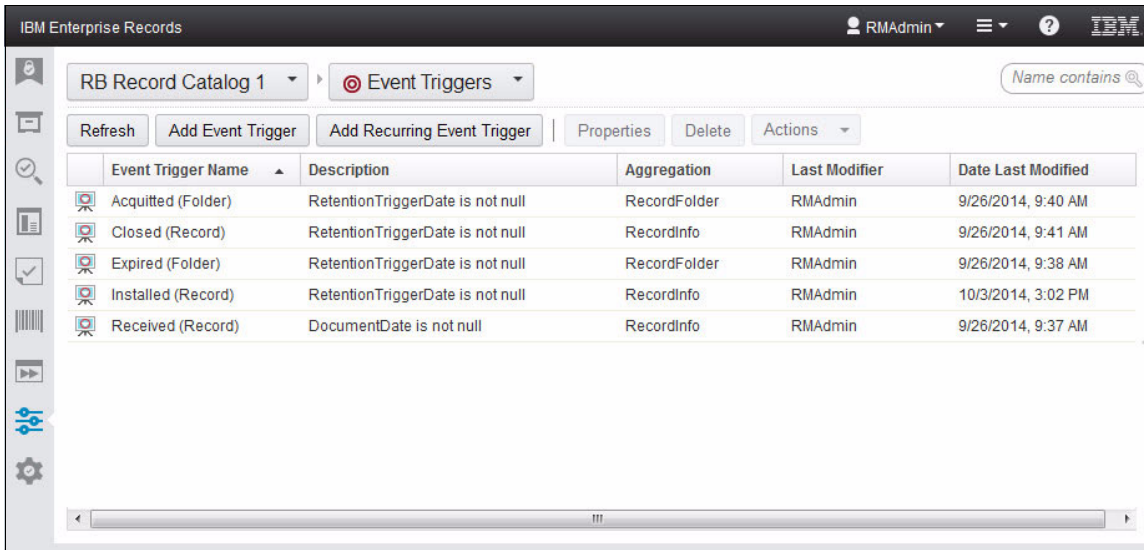
One important aspect of a trigger is the *aggregation* to which the trigger applies. This aggregation is especially important when defining an internal event trigger, because the properties available for the trigger condition depend on the aggregation selected. The following aggregation levels are available for internal event triggers:

- ▶ Record category
- ▶ Record folder
- ▶ Volume
- ▶ Record

Aggregation has a direct impact on the behavior of advanced disposition sweep. The aggregation selected for the trigger determines on which entities sweep operates. It also determines at which level entities are batched for disposition processing.

Figure 7-3 shows the disposal triggers that we have configured for the advanced disposition schedules in the sample file plan. The event triggers in this example are all internal event triggers. The event trigger name has a single word to describe the nature of the trigger and, in parenthesis, we also included the aggregation for reference. The description is populated with additional details related to the trigger condition, such as the symbolic name of the Date property that will be used to compute the retention and the conditions that will trigger retention.

In this example, all triggers related to event-based retention are using the same *RetentionTriggerDate* property and the one trigger related to fixed retention (that is, based on the document received date) is using the *DocumentDate* property. Also, notice that in this example, we have event triggers that use both *record* and *record folder* aggregation.



The screenshot shows the IBM Enterprise Records web interface. At the top, there's a header with 'IBM Enterprise Records', a user profile 'RMAAdmin', and navigation icons. Below the header, a breadcrumb trail shows 'RB Record Catalog 1' and 'Event Triggers'. A search bar on the right contains the text 'Name contains @'. Below the breadcrumb, there are buttons for 'Refresh', 'Add Event Trigger', 'Add Recurring Event Trigger', 'Properties', 'Delete', and an 'Actions' dropdown. The main content area is a table with the following data:

Event Trigger Name	Description	Aggregation	Last Modifier	Date Last Modified
Acquitted (Folder)	RetentionTriggerDate is not null	RecordFolder	RMAAdmin	9/26/2014, 9:40 AM
Closed (Record)	RetentionTriggerDate is not null	RecordInfo	RMAAdmin	9/26/2014, 9:41 AM
Expired (Folder)	RetentionTriggerDate is not null	RecordFolder	RMAAdmin	9/26/2014, 9:38 AM
Installed (Record)	RetentionTriggerDate is not null	RecordInfo	RMAAdmin	10/3/2014, 3:02 PM
Received (Record)	DocumentDate is not null	RecordInfo	RMAAdmin	9/26/2014, 9:37 AM

Figure 7-3 Disposal triggers

The internal event trigger

The internal event trigger has the following settings:

- Disposal trigger name
- Aggregation
- Description
- 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 *description* is useful for describing the trigger condition in more detail. The *condition* is a set of one or more properties used to satisfy the trigger condition.

The most common internal event trigger condition is a single relevant Date property not being empty (that is, being populated with a date value rather than being null). However, it is possible to include multiple properties to satisfy the trigger condition.

Figure 7-4 and Figure 7-5 on page 183 show an example of a common configuration for an event-based trigger condition using an internal trigger event. The trigger is configured for record aggregation. The Conditions tab shows the symbolic name of a single Date property and the comparison operator used to satisfy the condition, in this case, *Is Not Empty*.

Properties

Conditions

*Internal Event Name: ?

Closed (Record)

Internal Event Description: ?

RetentionTriggerDate is not null

*Aggregation: ?

Record

Event Trigger

An event is the occurrence of a specified condition based on which the system triggers an action on entities. An event is attached to a disposition schedule and automatically triggers the cut-off for the entity with which the schedule is associated. [Learn more](#)

Apply

Save

Cancel

Figure 7-4 Internal event trigger settings

The screenshot shows a configuration window with two tabs: 'Properties' and 'Conditions'. The 'Conditions' tab is active, displaying a list of conditions for the 'RetentionTriggerDate' property. The condition 'Is Not Empty' is selected. Below the list, there is an 'Add Property' button and two radio buttons: 'Any of the properties' (unselected) and 'All of the properties' (selected). To the right, a dark sidebar titled 'Event Trigger' contains explanatory text and buttons for 'Apply', 'Save', and 'Cancel'.

Figure 7-5 Internal event trigger condition typically includes a single Date property

This trigger can be assigned to one or more advanced disposition schedules that require such a condition. In our government case study and sample file plan, this trigger is used for the advanced disposition schedule for the 0300-150 record series.

7.1.3 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 of 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, a 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 unless the disposition schedule is changed. 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 a sweep. It is calculated based on cutoff base and delay.

Cutoff formula: Cutoff date = cutoff base + delay

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 the sweep detected the event and performed the disposition calculations for the entity. In most cases, you it is best to select a specific Date property to serve as the cutoff base and align this property with the same Date property that is used for the internal trigger event condition. For example, in our case study, the *RetentionTriggerDate* property is selected as the cutoff base for the 0500-150 disposition schedule because that schedule uses a trigger that depends on that Date property to satisfy the trigger condition. This approach removes any dependency on when we run disposition sweep. Whenever we decide to run the sweep, it accurately computes the appropriate cutoff date.

Delay

The *delay* is a time interval that gets added to the cutoff base to delay cutoff of an entity when a sweep computes the cutoff date. In certain configurations, the delay value can serve as the retention period when combined with a zero interval for the phases of disposition. This approach might be useful if trigger date values are expected to change up to the end of a retention period.

Cutoff action

The *cutoff action* is a specific disposition workflow that is automatically launched when an advanced 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 the cutoff is finalized, which provides an opportunity for a user to manually adjust the 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 desirable (for example, you might not want your users to be able to manually adjust the cutoff date). The *Cutoff Workflow* is provided with the software as a process that is associated with the cutoff action.

If there is no cutoff action specified for a disposition schedule, the sweep automatically computes the cutoff date. After the cutoff date occurs, the sweep also computes other phase information properties for the entity being disposed of that indicate cutoff has occurred.

Configuring cutoff in the disposition schedule

Cutoff is configured as part of the disposition schedule settings. Figure 7-6 shows the trigger and cutoff settings for the 0500-150 disposition schedule. The Cutoff base is set to the specific Date property, in this case, *RetentionTriggerDate*, that aligns with the selected trigger value. In this example, the Cutoff delay is set to zero and no cutoff action has been selected. With this configuration, when the trigger condition is satisfied and a sweep is run, cutoff is achieved immediately.

Properties Trigger and Cutoff Phases

Set the condition that triggers a cutoff of entities. Specify when cutoff will occur. By default, the cutoff base date is set to when disposition sweep is run.

* Trigger: ? Internal Event Trigger

* Trigger value: Closed (Record) × Select... Create

* Cutoff base: ? RetentionTriggerDate

Cutoff action: ? Select... Create

* Cutoff delay: ? 0 Years 0 Months 0 Days

Figure 7-6 Trigger and cutoff settings for an advanced disposition schedule

7.1.4 Disposition phases and actions

With advanced disposition schedules, the process for disposal of records is encapsulated in the phases of the disposition schedule. An advanced disposition schedule contains one or more disposition phases. Associated with each phase of disposition is a default retention period and 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 7-7 on page 186 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.

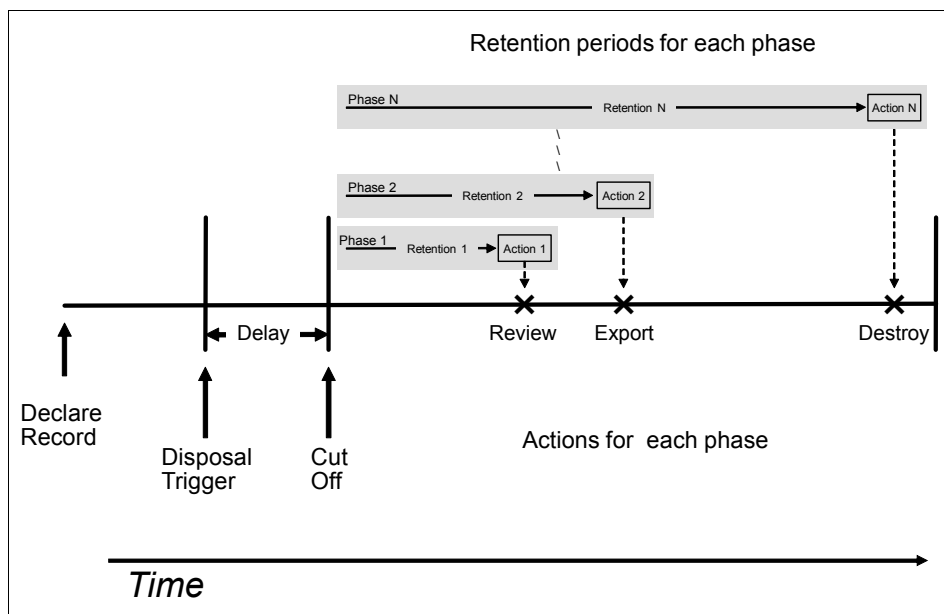


Figure 7-7 Retention periods and actions for each phase of disposition

While Enterprise Records allows a disposition schedule to be configured with multiple phases, many common use cases call for only one disposition phase that completes the destruction process. The number and type of phases are guided by your business requirements, your specific use cases, and your underlying records management policies and procedures.

Disposition actions

The *disposition action* defines what needs to happen to the entity being disposed of 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, Enterprise Records is tightly integrated with the underlying process capabilities of the IBM FileNet P8 Platform.

The following action types are used for disposition actions associated with the phases of disposition:

- ▶ Review
- ▶ Export
- ▶ Interim Transfer
- ▶ Transfer
- ▶ Destroy
- ▶ Auto Destroy

We describe each of these action types in more detail the subsections that follow and include information about the predefined workflows that are associated with each action type. The predefined workflows are provided with the software installation and can be used as is or customized to meet specific business requirements related to disposition processing.

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 software as a simple 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 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 simple 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 P8 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 IBM Enterprise Records, IBM Content Navigator, or any other application. So you cannot expect to retrieve the content from the P8 repository, 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 simple 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 simple 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 simple 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`. System performance and sizing should be taken into account when using the retain metadata option because the retained record metadata is not deleted automatically. A separate process and strategy should be considered for eventually removing the retained metadata or ensuring that the system can accommodate the continual growth without affecting performance.

Auto Destroy

The *Auto Destroy* action indicates that the records must be destroyed from the system immediately without relying on a configurable destruction process or user intervention. This action is implemented by an option included in an advanced disposition sweep and can be used for electronic records that do not require manual approval before destruction. Scheduling an *Auto Destroy Sweep* initiates and completes this action automatically.

There is no workflow associated with this action and there is no need to initiate disposition.

Ordering of multiple disposition phases

When adding multiple phases to a disposition schedule, the phases should be added in a logical order. 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 based on the action type. For example, the system prevents you from adding any disposition phases after a destruction or transfer phase.

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 of 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.

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.

The *Screening Workflow* is provided as a simple screening process.

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 configure a single screening workflow only *systemwide*.

7.1.5 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 for disposition. Because Enterprise Records is fully integrated with the underlying business process management capabilities of the FileNet P8 Content Platform Engine, you can customize these workflows to meet specific business requirements associated with the various disposition actions. The workflows included with the software (ready for use) are simple processes that provide basic functions. These workflows can be modified to provide customized disposition processes.

Here, we list the workflows that are provided with Enterprise Records for advanced disposition. These workflows are briefly described in the previous sections 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 these two:

- ▶ *Cutoff*: Associated with the cutoff action
- ▶ *Screening*: Associated with the screening option

7.1.6 Alternate retention

Alternate retention is a feature that allows you to add flexibility to your disposition schedules in situations where certain records that are being disposed of have similar disposition requirements, but their retention periods differ based on specific conditions associated with each record. For example, an 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 supersede 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 specifies one or more alternate retention periods.

Every disposition phase has a default retention period that 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 way to apply conditional retention to entities that are governed by the same disposition schedule. It overrides the default retention specified for a phase when certain conditions are met.

Alternate retention can be a useful feature for implementing conditional retention. However, it does introduce additional performance overhead and might not be the best option for certain scenarios or use cases. Solutions that require managing retention policies that varies by region, country, or jurisdiction typically implement different disposition schedules and record categories for such variations so that they can more be more effectively managed at the policy level.

7.1.7 Assigning disposition schedules to the file plan

File plans are hierarchical by nature. Their design reflects the classification schema that an organization uses to manage its records. Top-level nodes are typically broad groupings based on business functions. Subsequent levels of the file plan narrow the groupings to separate records according to these criteria:

- ▶ The retention policy that should apply to the records
- ▶ Who is responsible for and should have access to the records

When using advanced disposition schedules, you can organize the record categories in the file plan to separate your records by the retention and access requirements and assign the disposition schedules to the record categories to apply the retention policy. When assigning the disposition schedules to the record categories, the aggregation of the disposal trigger must match how records are organized within each record category.

One of the simplest ways to organize and assign disposition schedules is to define one disposition schedule for each record series. For each record category that represents the record series, you assign the disposition schedule that matches.

Figure 7-8 shows our case study file plan for government, in which each record category at Level 2 represents a record series. This example shows the Financial Management category selected and expanded to list the record series for that parent category. We use the record series code to uniquely identify each of these categories. In conjunction with defining the record categories, we also define one disposition schedule for each record series and use the record series code as part of the schedule name. Then, we assign each disposition schedule to the corresponding record category for each record series. This effectively applies the correct retention rule to all of the records that will be declared into each category.

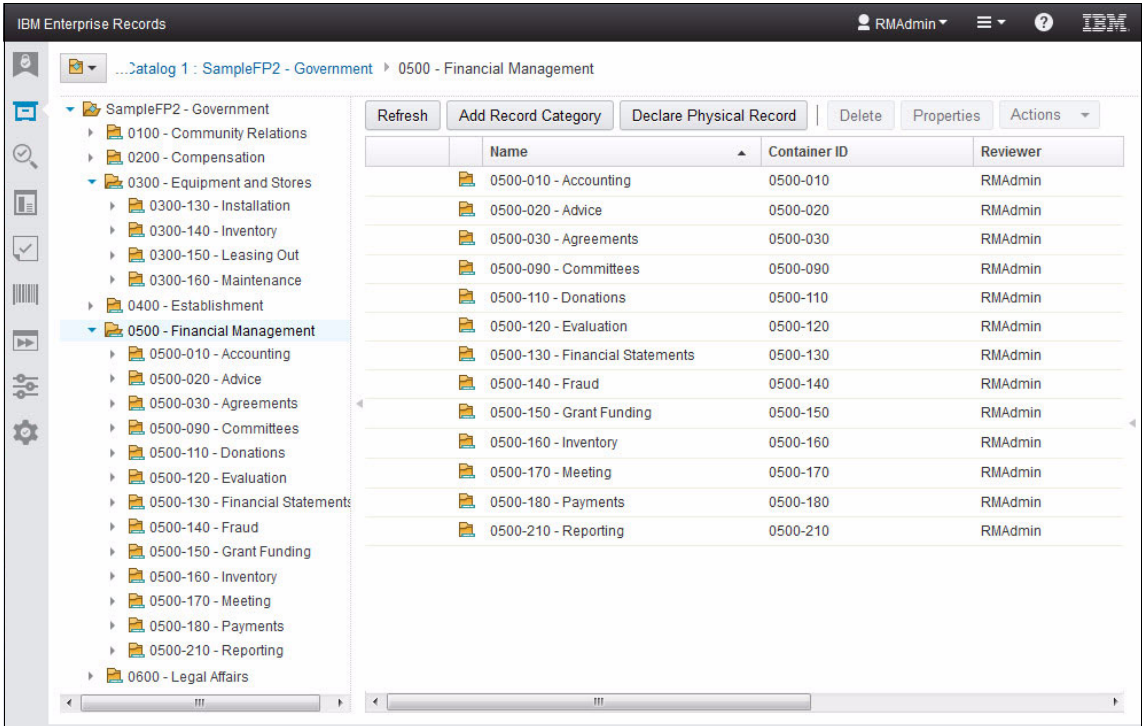


Figure 7-8 Browsing the government case study file plan

Figure 7-9 shows the list of disposition schedules for some of the categories across the file plan.

IBM Enterprise Records

RMAAdmin

?

IBM

RB Record Catalog 1

Advanced Disposition Schedules

Refresh

Add Disposition Schedule

Properties

Delete

Actions

	Schedule Name	Disposition Authority	Description	Last Modifier
⚙	0300-130 - Installed + 2Y	AFDA1149	Destroy 2 years after equipment is installed and configured	RMAAdmin
⚙	0300-150 - Closed + 7Y	AFDA1155	Destroy 7Y after lease expiration	RMAAdmin
⚙	0500-030 - Expiration + 7Y	AFDA1215	Destroy 7 years after expiration or termination	RMAAdmin
⚙	0500-010 - Received + 7Y	AFDA1219	Auto Destroy 7 years after document is received	RMAAdmin
⚙	0500-170 - Received + 7Y	AFDA1239	Destroy 7 years after document is received	RMAAdmin
⚙	0500-130 - Received + 7Y	AFDA1250	Destroy 7 years after the document is received	RMAAdmin
⚙	0500-150 - Acquitted + 7Y	AFDA1253	Destroy 7 years after grant acquittal	RMAAdmin
⚙	0500-160 - Received + 2Y	AFDA1254	Auto Destroy 2 years after document is received	RMAAdmin

Figure 7-9 List of advanced disposition schedules

Figure 7-10 on page 194 is an example search result display that shows how each advanced disposition schedule is assigned to the record categories according to the matching record series code. By using the record series code to name each disposition schedule, we can easily assign the correct schedule to each record category for each record series.

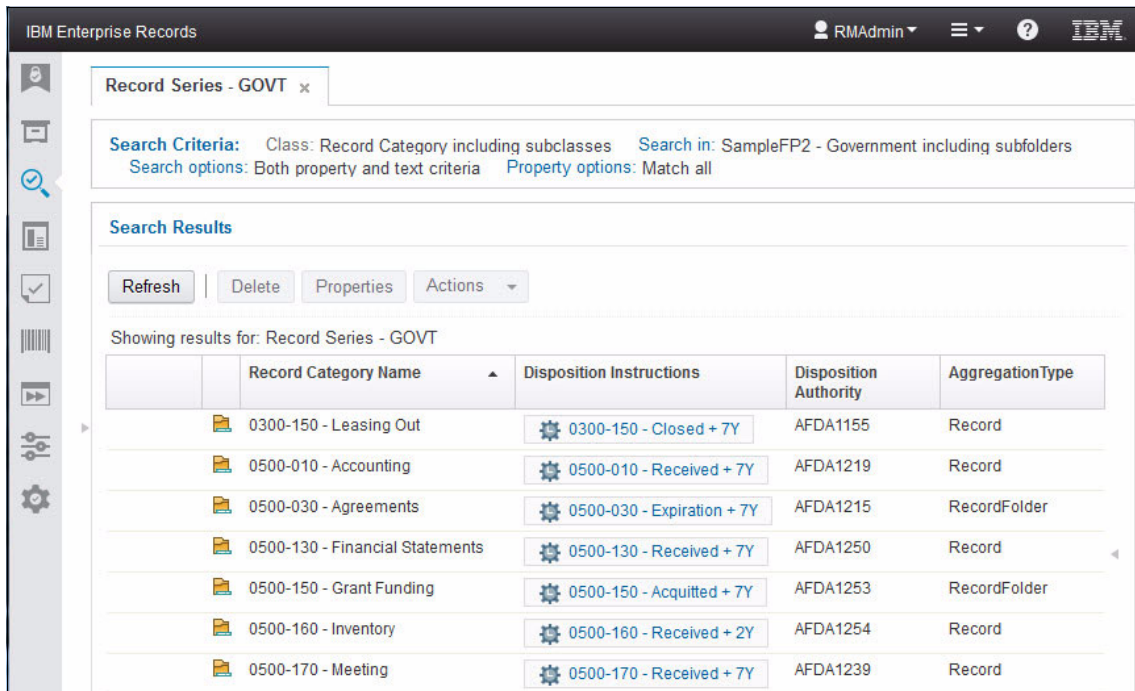


Figure 7-10 Search results showing how disposition schedules are assigned to record categories

The roles of inheritance and aggregation

In a typical file plan, records (or other entities being disposed of, such as record 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. Advanced disposition schedules are associated 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.

Overriding a parent schedule by applying a different schedule at a lower level can be a useful technique to quickly deal with exceptions to retention policy. However, it is a more common practice to design the file plan so that variations to retention policy are clearly accounted for in the way record categories are organized to represent record series and their exceptions.

Figure 7-11 shows how our case study sample file plan for government is organized. Record categories at the second level of the file plan represent each of the record series for a given function and the corresponding advanced disposition schedule is assigned to each of these record categories. Some of the record series in this sample use record-level aggregation and some use record folder-level aggregation. In this sample file plan, we chose to create one disposition schedule for each record series so we could easily apply the disposition schedule to the appropriate record category.

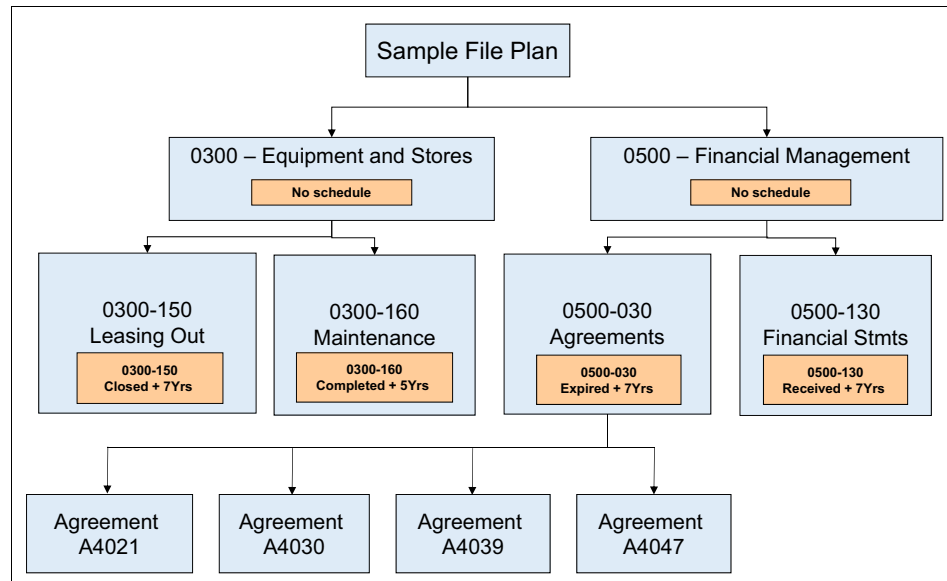


Figure 7-11 Organizing a file plan for disposition schedule inheritance

The “0500-030 - Agreements” record series illustrates folder-level aggregation, where all records are organized into record folders: One folder for each agreement based on a unique agreement number. The disposal trigger for this disposition schedule is configured with folder-level aggregation to align with the use of record folders. When the advanced disposition sweep is run on this record category, the sweep looks for record folders that are ready for disposition. It does not look at individual records. The disposition process disposes of each record folder and the records in that folder.

Figure 7-12 on page 196 shows how, in this example, record folders are created for each agreement that is to be managed. Agreement records are not declared directly in the record category but are declared for the specific record folder that matches the agreement number.

The disposal trigger is the Date value set for each record folder when that particular agreement has expired or been closed. When each record folder is due for disposition, all of the records in that folder are processed along with the folder.

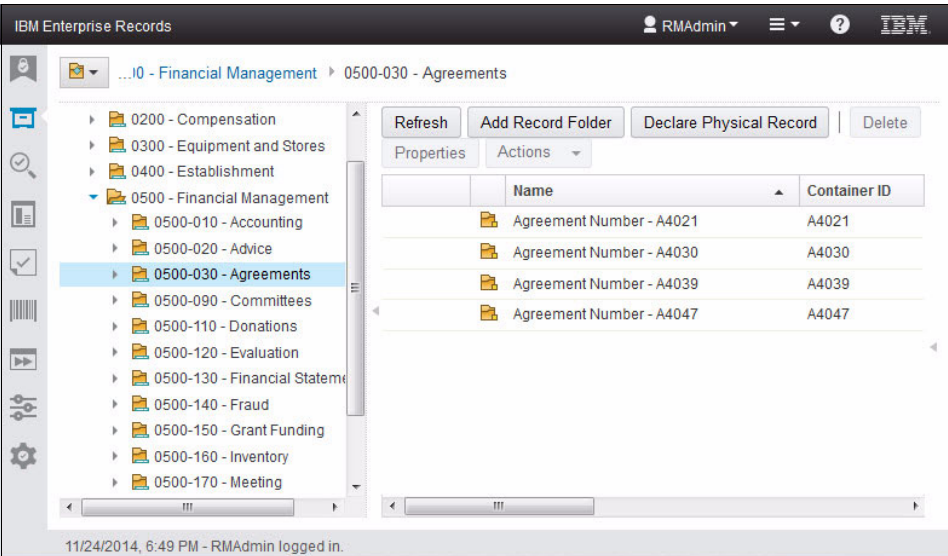


Figure 7-12 Agreements organized by record folders, one for each agreement number

In contrast, the “0500-130 - Financial Statements” record series illustrates record-level aggregation, where all records belonging to a given record series are declared in the record category directly without the need for record folders. This is a common solution approach for fixed retention where the disposal trigger is based on the date of each document and is known and set at the time of declaration.

Figure 7-13 on page 197 shows how records are declared directly into the 0500-130 category because that category has been configured for record-level aggregation. When using record-level aggregation, there is no need for record folders.

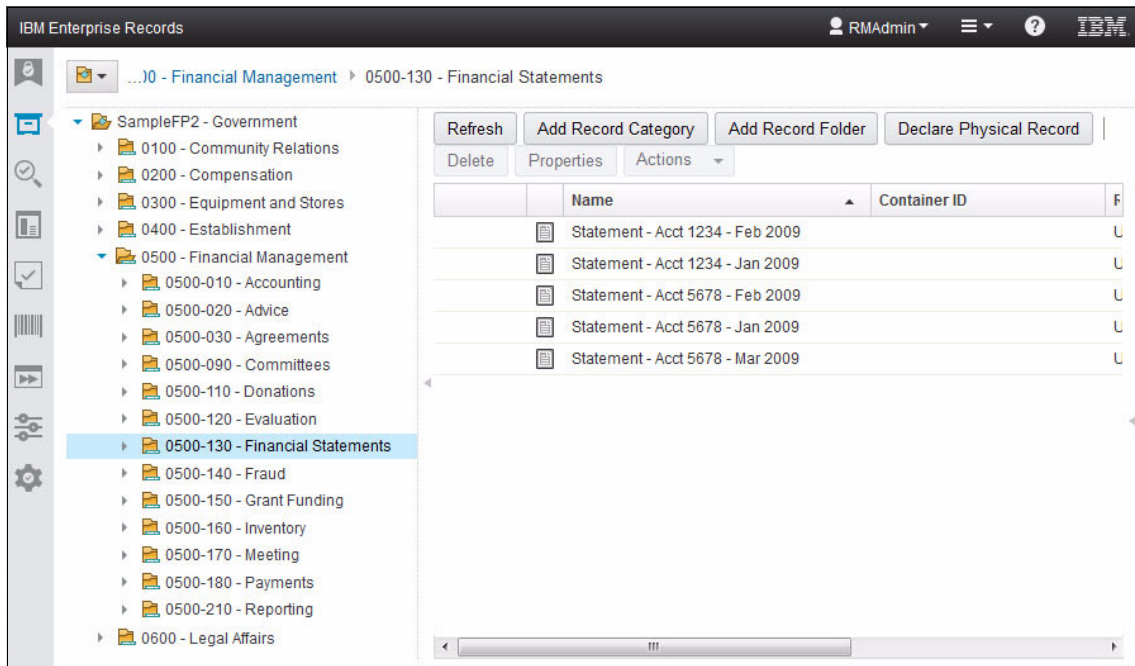


Figure 7-13 Financial statements are declared directly into the 0500-130 record category

There are tradeoffs between these two approaches:

- ▶ Folder-level aggregation requires that a process be put in place to create individual record folders. However, the advanced disposition sweep will run more efficiently because it must encompass only compute values for each folder, rather than for each record.
- ▶ Record-level aggregation does not require the creation or management of record folders, but it requires more effort for the advanced disposition sweep to compute the retention for each record individually.

Deciding which of these options to use for a given record series is a design decision that is determined by the solution requirements and system performance considerations.

For more examples and information, see the *Best practices to improve performance for IBM Enterprise Records* Technote:

<http://ibm.co/1E2IqvB>

7.1.8 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) even though the disposition requirements are still enforced based on the type of record. In this special case, there is a feature in Enterprise Records 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 effect, 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 use: 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, it is a best practice to 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 record folders in each category according to their type. This design enables you to take advantage of assigning the disposition schedule at the category level so that it can be inherited by the child records that are ready for disposition.

Caution: Use the *Record Types* feature only 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. Rather than using record types, it is preferable to design your file plan so that records are grouped according to disposition requirements. Also, there is extra processing effort associated with the use of record types, because a sweep must perform more work when calculating disposition parameters for record types. So use *Record Types* only when the requirements demand using this feature.

7.2 Advanced disposition sweep

An advanced disposition sweep is used to process records for disposition when using advanced disposition schedules. The sweep has three separate functions and works a bit differently from the basic disposition sweep. Advanced disposition involves scheduling two separate functions, one after the other. First, you must run the disposition sweep for *disposition processing*. Then, you must either *Initiate Disposition* to launch a workflow or run the sweep for *Auto Destroy*.

This section focuses on using the advanced disposition sweep for *disposition processing*. Using the advanced disposition sweep for initiating disposition and for Auto Destroy are described in subsequent sections.

7.2.1 Advanced disposition sweep for disposition processing

An advanced disposition sweep computes essential disposition parameters to enable disposition processing. It is typically scheduled to run periodically to analyze a file plan and determine which entities are ready to be processed. The disposition sweep must complete before you can either initiate disposition or run the sweep for Auto Destroy.

The advanced disposition sweep performs the following functions automatically, depending on how the disposition schedules and sweep have been configured:

- ▶ Determines entities eligible for disposition
- ▶ Calculates cutoff
- ▶ Performs phase transition
- ▶ Performs phase updates
- ▶ Initiates the cutoff action
- ▶ Handles housekeeping tasks (record maintenance)
- ▶ Processes vital records

Determining entities that are eligible for disposition

A disposition sweep uses the aggregation specified in the disposition schedule configuration to determine which entities are eligible for disposition processing. The 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

The disposition sweep calculates the cutoff date for entities that are ready for disposition. After the condition for the cutoff trigger is satisfied, the disposition sweep calculates the cutoff date by using either the *event date* or the cutoff base and the delay. For more information, more information “Computing cutoff” on page 184 and “Cutoff base” on page 184.

Performing phase transitions

The disposition sweep moves record entities from one phase to the next phase. In the case where the phase has no associated action, the disposition sweep automatically promotes the entity to the next phase. When there is an action associated with a phase, the disposition sweep relies on the underlying workflow to move the entity to the next phase when the workflow completes.

Performing phase updates

A disposition sweep detects whether the phases of an entity’s disposition schedule have been changed (for example, the retention period or action have been 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, the disposition sweep automatically launches the associated workflow when a record entity reaches cutoff.

Housekeeping (updates)

The disposition sweep performs several internal housekeeping or maintenance chores. One task involves 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, the disposition sweep detects this condition and updates the affected entities accordingly.

Processing vital records

The 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.

7.2.2 Setting an advanced disposition sweep to run from the desktop

Before you run an advanced disposition sweep from the desktop for the first time, verify the sweep connection type and create a folder for sweep logs.

Sweep connection type

RecordsManagerSweep must be configured for EJB (Enterprise JavaBeans) mode. The RecordsManagerSweep.bat or RecordsManagerSweep.sh file must set CONNECTION_TYPE to EJB. The Enterprise Records desktop supports only CONNECTION_TYPE=EJB. This parameter is usually configured during installation.

Logs output directory

Create or identify a repository folder where you want the sweep logs output to be stored. The account used to run the sweep must have permission to file documents in that folder. The logs can be stored on any FileNet P8 repository that is available to the Enterprise Records desktop.

7.2.3 Running an advanced disposition sweep from the desktop

To run an advanced disposition sweep for *disposition processing*, you can specify parameters for the sweep and then set the schedule for the sweep from the Enterprise Records desktop Tasks view.

Set the sweep parameters

The following parameters are set when scheduling an advanced disposition sweep for disposition:

- ▶ *File plan repository*: The file plan object store (FPOS) you want to sweep.
- ▶ *Containers for sweep*: You must select one or more containers to sweep.
- ▶ *Logs output directory*: The repository folder where log files will be stored.
- ▶ *Type*: The type must be set to **Disposition**.

Additional parameters can also be adjusted when scheduling the sweep for disposition:

- ▶ *Workflow connection point*: The connection point to use for workflow processing, which is usually not changed after it is initially configured
- ▶ *Run for record types*: Default is No, use only when needed
- ▶ *Run for vital*: Default is No, use only when needed

Set the sweep run schedule

The disposition sweep has the following set of options for setting the sweep run schedule:

- ▶ *Name*. You must provide a name for the sweep task that helps identify the sweep task in the Task view.
- ▶ *Description*. You can include a more detailed description that explains the nature of the sweep task.
- ▶ *Schedule*. You can select whether to run once or run on a schedule.
 - *Run once*. You can specify a start time or choose to start immediately.
 - *Run on a schedule*. You must specify the frequency, a start date and time, and, optionally, an end date.
- ▶ *Login information*. For an advanced disposition sweep, you must provide valid login credentials. The login credentials should have sufficient access to process and update the entities that are being swept.
- ▶ *Notification*. You can provide an email address for notification (optional).

After the advanced disposition sweep has completed, you can either initiate disposition for those disposition schedules that rely on workflow for disposition processing, or you can run the auto destroy sweep to complete the auto destroy process for those schedules configured for automatic destruction.

The sweep run is limited to the portion of the file plan that is selected by the *Containers for sweep* parameter. Because a file plan is hierarchical, the sweep will include all entities contained within the selected containers. For example, you can select the entire file plan as the single container to sweep, or you can select one or more specific record categories to sweep.

7.3 Initiating and completing disposition

When using disposition schedules that involve workflow, after the advanced disposition sweep has completed, you must initiate the disposition process for the records that are ready for disposition. Initiating disposition can be done manually by selecting entities and issuing the **initiate disposition** command, or you can schedule a sweep to initiate disposition.

Initiating disposition launches the workflows that implement the disposition actions. The workflows define the steps required to complete the disposition actions.

This list summarizes the events that lead 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).
- ▶ A 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, so the action associated with the disposition phase is ready to be initiated.

Note: For a sweep to run from the desktop, the sweep must have been configured to run using EJB mode.

7.3.1 Initiating disposition manually

You can initiate disposition directly on the sections of the file plan by selecting what you want (typically a record category at a higher level in the file plan) and issuing the command. 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, 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 throughout 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, which enables the Initiate Disposition function to automatically aggregate entities in batches.

The Initiate Disposition function ignores any entity that is not ready for disposition. This includes any entity that is not configured for disposition, any entity for which the retention period has not yet ended, or any entity on hold.

7.3.2 Initiating a disposition by scheduling a sweep

You can initiate disposition automatically by scheduling a sweep from the Enterprise Records desktop Task view.

Set the sweep parameters

The following parameters are set when scheduling the sweep:

File plan repository	The file plan object store (FPOS) you want to sweep.
Containers for sweep	You must select one or more containers to sweep.
Logs output directory	The repository folder where log files will be stored.
Type	The type must be set to Initiate Disposition.
Workflow connection point	The connection point to use for workflow processing.

Set the sweep run schedule

The disposition sweep has the following set of options for setting the sweep run schedule:

- ▶ *Name*. You must provide a name for the sweep task that will be used to help identify the sweep task in the Task view.
- ▶ *Description*. You can include a more detailed description explaining the nature of the sweep task.
- ▶ *Schedule*. You can select whether to run once or run on a schedule.
 - *Run once*. You can specify a start time or choose to start immediately.
 - *Run on a schedule*. You must specify the frequency, a start date and time, and optionally an end date.
- ▶ *Login information*. For an advanced disposition sweep, you must provide valid login credentials. The login credentials should have sufficient access to process and update the entities being swept.
- ▶ *Notification*. You have the option to provide an email address for notification.

7.3.3 Strategies for initiating disposition

Whether to initiate across the entire file plan in a single run or to limit the initiate disposition to selected record categories depends on the size of the file plan, on the number of entities expected to be ready for disposition, on the performance

capacity of the system, and on the specific solution requirements at any given time. For most production systems that have millions of records, a strategy should be developed to accommodate these factors. The practical approach taken depends on the file plan design and your own business practices and records management policies and procedures.

Preferable practice: Records Managers must develop a strategy and plan for 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 a 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, use this approach only in special circumstances. It is typically a records management best practice to routinely initiate disposition across selected categories, which allows the system to aggregate entities into batches automatically.

7.3.4 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 Enterprise Records enables you to configure the batch size for disposition workflows. This parameter is a systemwide 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 determined primarily by your solution 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 criteria to group entities into separate batches:

Reviewer	The Reviewer property value that is assigned to the record that is ready for disposition
Action	The specific action that is assigned to the disposition phase that is being initiated

The Reviewer property is a required property for all record entities. By default, the Reviewer property is 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 to control how entities are batched. A single batch includes only entities that are assigned the same reviewer.

Useful technique: 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. Even though the Reviewer property is limited to a valid user ID when using the Enterprise Records desktop to set this value, it is actually stored as a string property that can programmatically be set to any string value. You can use this feature to your advantage to set this string to a more meaningful value that can indicate which area of the business is responsible for reviewing or processing these records. This technique can be more useful than assigning the Reviewer property to a specific user who might not even be in the organization when the time comes to process disposition.

The action that you assign to a disposition phase is also used to control the batch groupings. A single batch includes only entities that are assigned the same action. You can use this feature to your advantage to control batching, to a degree, by either configuring separate actions for different disposition schedules or by reusing the same action across multiple disposition schedules.

7.3.5 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. 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 7.1.4, “Disposition phases and actions” on page 185. You can customize the predefined workflows that are provided to meet your specific solution requirements.

When disposition schedules are configured with multiple phases, disposition must be initiated separately for each phase. After the disposition process for the first phase completes, you must schedule a sweep to initiate and complete each remaining phase.

7.4 Automatic destruction using Auto Destroy

For disposition schedules configured with the Auto Destroy action, you must run the Auto Destroy sweep to complete the destruction of records. The sweep should be run after an advanced disposition sweep has completed and you are ready to have the records immediately destroyed.

7.4.1 When to use Auto Destroy

Auto Destroy is implemented for use cases where no approval is required before record destruction and where there is a large volume of records being managed and destroyed by the system. However, with the introduction of basic disposition schedules and the basic disposition process, Auto Destroy might not be the optimal choice for such use cases. You can achieve automatic destruction more efficiently when you use basic disposition, depending on your solution requirements and the design of your file plan.

Auto Destroy is compatible only with advanced disposition schedules and must be used when you have advanced disposition schedules configured with the Auto Destroy action.

7.4.2 Running Auto Destroy from the desktop

You can run Auto Destroy by scheduling a sweep from the Enterprise Records desktop Task view.

Set the sweep parameters

The following parameters are set when scheduling the sweep:

- | | |
|-----------------------------|---|
| File plan repository | The file plan object store (FPOS) that you want to sweep. |
| Containers for sweep | You must select one or more containers to sweep. |

Logs output directory	The repository folder where log files will be stored.
Type	The type must be set to Auto Destroy.
Generate transcript	Default is Yes, but it can be set to No.

Set the sweep run schedule

The disposition sweep has the following set of options for setting the sweep run schedule:

- ▶ *Name.* You must provide a name for the sweep task that will be used to help identify the sweep task in the task view.
- ▶ *Description.* You can include a more detailed description explaining the nature of the sweep task.
- ▶ *Schedule.* You can select whether to run once or run on a schedule.
 - *Run once.* You can specify a start time or choose to start immediately.
 - *Run on a schedule.* You must specify the frequency, a start date and time, and optionally an end date.
- ▶ *Login information:* For an advanced disposition sweep, you must provide valid login credentials. The login credentials should have sufficient access to process and update the entities being swept.
- ▶ *Notification.* You can optionally provide an email address for notification.

An advanced disposition sweep should be run before the Auto Destroy sweep runs. This is to allow the disposition date values to be computed, which makes records ready for disposition at the appropriate time.

7.5 Running a sweep from the command line

With the introduction of the sweep scheduling feature now available in the Enterprise Records desktop, most solutions typically use that feature to schedule and run disposition sweeps. However, the existing command-line RecordsManagerSweep tool is still available for situations when it is better to use the command line for running disposition sweeps.

7.5.1 Configuring an advanced disposition sweep

If you plan to use the advanced disposition sweep as a command-line tool, you must configure the tool on the server where the Enterprise Records tools are installed. The command-line tool is typically installed on the IBM FileNet P8 Content Platform Engine platform Content Engine server.

There are several parameters that must be specified. Although all of those parameters are important, we describe the following parameters in more detail:

File Plan Object Store Name	Name of file plan object store on which you want to run the sweep. If this parameter is left blank, all object stores in the domain that are configured to be an FPOS will be processed.
Container GUIDs	Identifies one or more nodes in the file plan that limits the scope of the sweep run. When a container Globally Unique Identifier (GUID) is entered, a 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 the sweep whether to include record types processing.
Run for Vital	Boolean flag that tells the sweep whether to include vital records processing.

For a full description of the features and configurable parameters, see “Configuring disposition sweeps” in IBM Knowledge Center:

<http://ibm.co/1PiXQCj>

The *File Plan Object Store Name* and the *Container GUIDs* parameters are useful when deploying an advanced disposition sweep to limit the scope of each run. The two parameters can be used in conjunction to limit the sweep to a particular file plan in a deployment that has multiple file plans, or they can be used to limit the sweep to a particular segment of a file plan.

Both *Run for Record Types* and *Run for Vital* options enable additional processing for a disposition sweep, so they add performance overhead when they are set to True. Record types and vital records are two features that are not frequently used.

Preferable practice: For the best performance, do not enable *Run for Record Types* and *Run for Vital* unless you use these features. When not using these features, ensure that these parameters are set to False.

7.5.2 Deployment and scheduling considerations

Typically, the advanced disposition sweep command-line tool is installed on the IBM FileNet P8 Content Platform Engine server, because having a sweep local to the Content Platform Engine can have performance benefits. However, a sweep can be installed on a completely separate machine if necessary, especially if the sweep must compete for limited resources on the Content Platform Engine.

There are several factors that can influence how a sweep is deployed in your environment:

- ▶ File plan size and design
- ▶ Platform topology
- ▶ Server capacity
- ▶ Performance considerations
- ▶ Records management policies and procedures
- ▶ Other processes that compete for resources on the Content Platform Engine

File plan size, design, and aggregation levels influence the number of entities that a 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 the sweep needs to run also strongly influence how the sweep should be deployed, configured, and used.

It is possible to deploy multiple instances of an advanced 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 a sweep and whether a single instance suffices or whether multiple instances are needed, consider these factors:

- ▶ *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 sweep schedules.
- ▶ *Administration.* Multiple file plans or segments within the same file plan can be administered by different Records Administrators.

Regardless of how it is run, scheduling of sweeps is an important consideration for any enterprise. There are several important factors to consider when scheduling sweeps:

- ▶ An organization needs to determine the frequency with which to run a 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, a disposition sweep can have a substantial impact on system performance while running, depending on the number of entities that a sweep needs to process. Therefore, it is important to coordinate running a disposition sweep with other enterprise activities, such as system backups, so that it is run during periods where system use is low and the impact on users is minimal.

An advanced 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 the sweep is performing and can be used to fine-tune the configuration for optimal performance.

7.6 Performance considerations

The file plan design, the number of records expected, and how you choose to aggregate for purposes of disposition have a direct impact on the performance of an advanced disposition sweep. Aggregation is an extremely influential factor in determining how many entities the sweep needs to process when it is run.

In general, the more often that you can aggregate at the container level (either record folder or volume), the fewer items the sweep must process. However, there are use cases and scenarios where it is impractical to aggregate at the container level. When aggregating at the record level, be sure you develop the appropriate strategies for deploying and running an advanced disposition sweep to handle the anticipated load.

There are also general performance tuning practices to follow for Enterprise Records. For example, by creating the appropriate indexes for system and custom properties in the database, you can greatly improve sweep performance. The specific properties that require indexing depend upon the metadata that you choose for your disposal trigger criteria. See the IBM publication titled *IBM FileNet P8 Performance Tuning Guide*, GC31-5483, for more information.

Preferable practice: Work with your database administrator (DBA) in the early stages of implementation to properly tune the database for Enterprise Records use. Become familiar with various performance tuning characteristics that are affected by your file plan design and your specific solution requirements.

7.7 Converting advanced schedules to basic schedules

Record categories that have been initially configured with either no disposition schedule or with an advanced disposition schedule can be converted to have a basic disposition schedule. This is a one-time conversion process that cannot be reversed. After a record category has been configured with a basic disposition schedule, it cannot be converted back to a record category with advanced disposition.

For more information, see 6.3.3, “Converting a record category to a basic schedule” on page 161.



Holds and preservation

To be prepared in the event of litigation, investigations, or audits, 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 ensure 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 describe hold processing in IBM Enterprise Records.

We cover the following topics in this chapter:

- ▶ Definition of hold
- ▶ Hold processing in IBM Enterprise Records
- ▶ Performance considerations

8.1 Definition of hold

A *hold* is an action taken on records or containers to stop the disposition process that is defined in the retention policy for that record to ensure 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.

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 geographical regions. Primary regulators in the US 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 of 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*.

8.2 Hold processing in IBM Enterprise Records

Enterprise Records 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 can be automated through the use of conditional (dynamic) holds. By default, only users and groups assigned the role of either Records Administrator or Records Manager can create and apply holds.

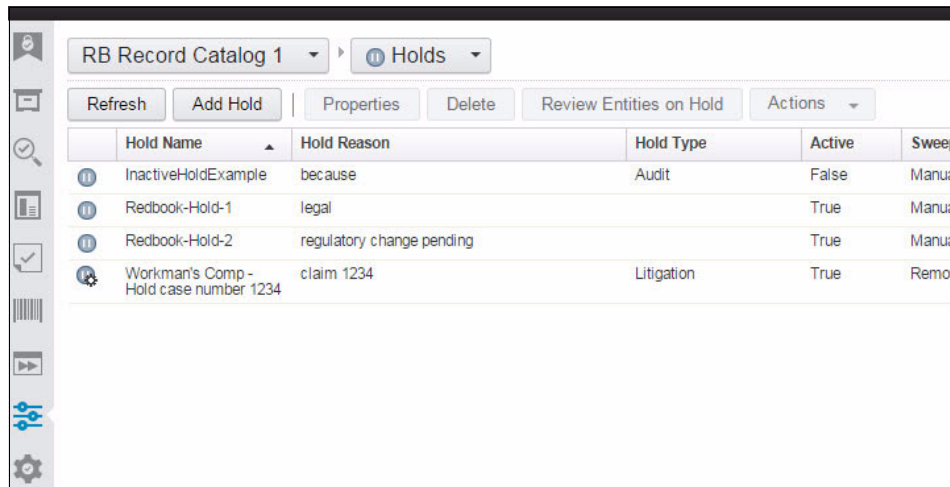
Holds are automatically inherited by lower levels of the file plan. Using our case study as an example, if a hold is placed on the Operations record category, all categories, folders, volumes, and records under Operations are automatically on hold. Enterprise Records prevents anyone from overriding this hold until the hold is removed.

When a hold is placed on an entity, the hold overrides normal disposition processing.

Enterprise Records provides two disposition processes:

- ▶ *Basic disposition process.* A record on hold will not be added to a basic disposition report. During the report review process, if a record listed in the report should not be deleted, it can be put on hold to prevent the basic disposition process from destroying it. Alternatively, the content associated with a record can also be put on hold by using eDiscovery Manager or IBM Content Process Engine. Because the content associated with a record cannot be destroyed, the record object will not be destroyed either. When an document associated with a record is on hold by other features, such as eDiscovery Manager hold, Content Process Engine hold, or Content Process Engine retention, the disposition process does not recognize that state, and the record will be added to the report for destruction. However, the destruction of the record will eventually fail because that associated document is on hold.
- ▶ *Advanced disposition process.* The hold prevents an authorized user from initiating disposition on the entity until the hold is removed. All disposition actions are suspended until the hold is removed. Therefore, if a record is placed on hold after it was added to a Destroy workflow, the workflow will not delete that record. The record will be eligible for disposition when the hold is removed. Putting a hold on an entity does not prevent disposition sweep from calculating the *Cutoff 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 must specify whether it is Active or Inactive (see Figure 8-1 on page 216). Only Active holds can be placed on records. A Records Administrator can define a hold at any time. However, if the hold is not to be used immediately, but later instead, or a Records Administrator determines that a hold is no longer to be used, the hold is set to Inactive.



Hold Name	Hold Reason	Hold Type	Active	Sweep
InactiveHoldExample	because	Audit	False	Manual
Redbook-Hold-1	legal		True	Manual
Redbook-Hold-2	regulatory change pending		True	Manual
Workman's Comp - Hold case number 1234	claim 1234	Litigation	True	Remove

Figure 8-1 Active and inactive holds

8.2.1 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 that are the subject of pending or potential *litigation or investigation* are not destroyed until the litigation or investigation is over and 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, so Enterprise Records 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 called HoldTypeList in the FPOS. To update these values, edit the choice list using Enterprise Manager.

8.2.2 Manual holds

A hold created with no conditions is a *manual hold*. A manual hold is designed for dynamic use when the Records Administrator or Records Manager search or browse for records to include in the hold. The manager manually puts these records on hold. Typically, manual holds are used only 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.

8.2.3 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 those records are potentially distributed throughout the file plan. A dynamic hold can also be run regularly 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 be added to an entity only 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 schedule, and that process automatically places holds on the new records that meet the hold conditions.

The search conditions are created by using the metadata on the entity (see Figure 8-2 on page 218). The search conditions can differ for each type of entity (category, folder, volume, or record) and can use multiple pieces of metadata per item, which are joined through the logical operators AND or OR. Retrieval of the content associated with a record can be used 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, the search does not search for records with document title that contain the words *ABank* and *statement*. Also, this condition applies to a record category that is created after 01/01/2015:

- For records with Document Title like "ABank%statement"
- For record categories with Date Created <= 01/01/2015

Instead, this search is for all records with Document Title that contain the words *ABank* and *statement*, and all record categories that are created on or before 01/01/2015.

Properties

*Hold Name: ?

ABank statements

Hold Reason: ?

Need to review ABank statements

Hold Type: ?

Audit

Active: ?

True

Conditions

Record (0)

Record Category (0)

Record Folder (0)

Record Volume (0)

Document Title

Like

ABank%Statement

Add Property

☐ Any of the properties

☒ All of the properties

Content Contains

☐ Any of the terms

☒ All of the terms

Preview

Preview Results

	Document Title	Last Modifier	Date Last Modified
	ABank February 2014 Bank Statement	RMUser	10/3/2014, 9:26 AM
	ABank March 2014 Bank Statement	RMUser	10/3/2014, 9:27 AM

Figure 8-2 Building search conditions for a dynamic hold

8.2.4 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. This can result 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 about obtaining discoverable records. This action automatically triggers Hold 1 on a record (see Figure 8-3 on page 219).

In this case, we guarantee that records will not be destroyed until the end of trial process 1.

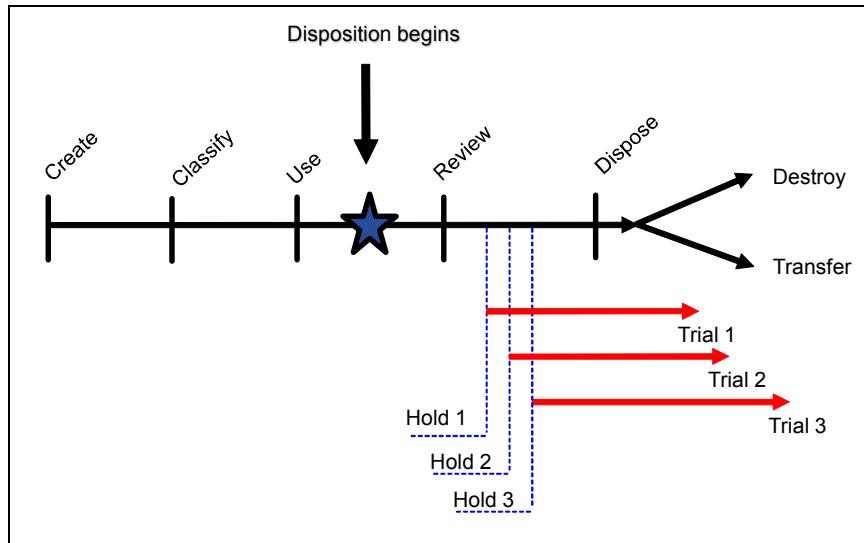


Figure 8-3 Example of multiple holds needed

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.

8.2.5 Applying holds

You can apply holds manually or automatically.

Placing a hold manually

By using the Browse page (Figure 8-4 on page 220), a Records Manager can navigate the file plan to identify entities to be placed on hold. Alternatively, the Search page enables the Records Manager to search for specific entities that need to be placed on hold.

To place an entity on hold manually, select the appropriate entity (or entities if you select multiple lines), right-click, and select the **Place On Hold** option from pop-up the menu.

Note: Any hold can be placed manually, even if it has a condition defined for hold sweep. When a condition hold is placed manually on an entity, it can be removed manually or by running a hold sweep for Initiate Remove Hold Request.

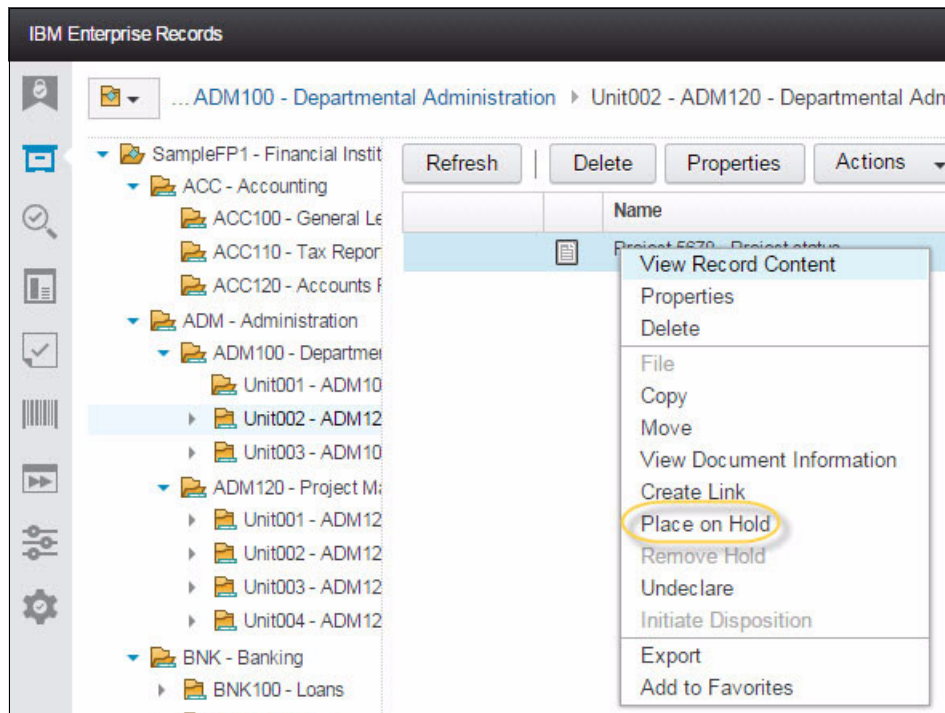


Figure 8-4 Browse page: Manually put records on hold

Placing a hold automatically

Holds can be placed on records automatically by using the Hold Sweep program. See 8.2.7, “Running Hold Sweep” on page 225, for details.

Hold icon

Whether a hold was placed manually or automatically, the hold icon is displayed next to the entity on hold, as shown in Figure 8-5 on page 221.

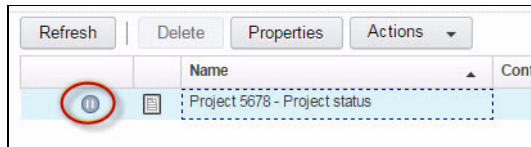


Figure 8-5 Icon showing that the record is on hold

8.2.6 Removing holds

Similar to placing holds on entities, you can remove holds from entities manually or automatically.

Note: Only holds placed manually can be manually removed. Holds placed by a hold sweep can be removed only by a hold sweep.

Removing holds manually

To manually remove holds, Enterprise Records offers two mechanisms:

- Browse the file plan by using the Browse page or search the held records directly by using the Search page.

To search for the held records directly, you can search for entities where the On Hold property value is equal to True.

After the appropriate record has been identified and selected, right-click and select the **Remove Hold** option from the pop-up menu, as shown in Figure 8-6 on page 222.

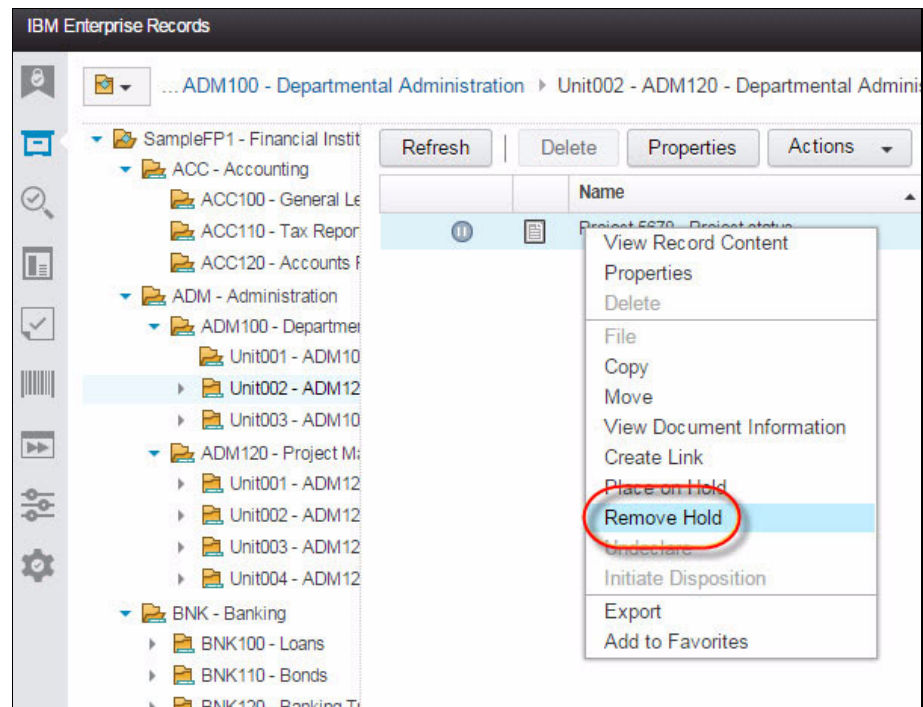


Figure 8-6 Selecting the Remove Hold option

Using this option, you are not allowed to remove a hold from multiple records at the same time.

After you select the Remove Hold option, you are asked to specify which hold you want to remove, as shown in Figure 8-7.

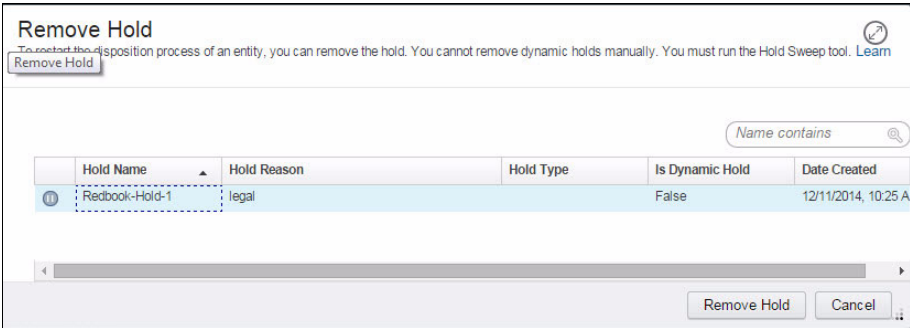


Figure 8-7 Selecting the hold to remove

- Use the Review Entities On Hold from the hold definition.
The most likely scenario is that the litigation or audit is complete and the associated hold must be removed from all records that are related to the specific litigation or audit. To remove this hold, use the **Review Entities On Hold** option that is available on the Admin page of the hold object, as shown in Figure 8-8.
- a. Select the **Configuration View** icon in the launch bar.
- b. Select **Holds** from the drop-down menu.
- c. From the Hold list, select the hold from which you want to remove the records, and select the **Review Entities On Hold** option (either through right-clicking and using the pop-up menu option or using the Actions button).

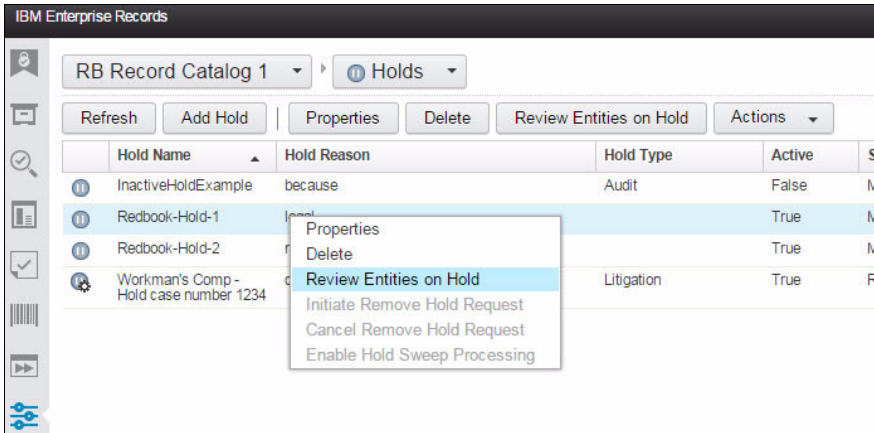


Figure 8-8 Review the entities on hold

- d. Search for the records that you want to remove from this hold. To list all records, leave the selection text box empty, and click **Search**.
- e. Select the records to remove, and click **Remove Hold**, as shown in Figure 8-9.

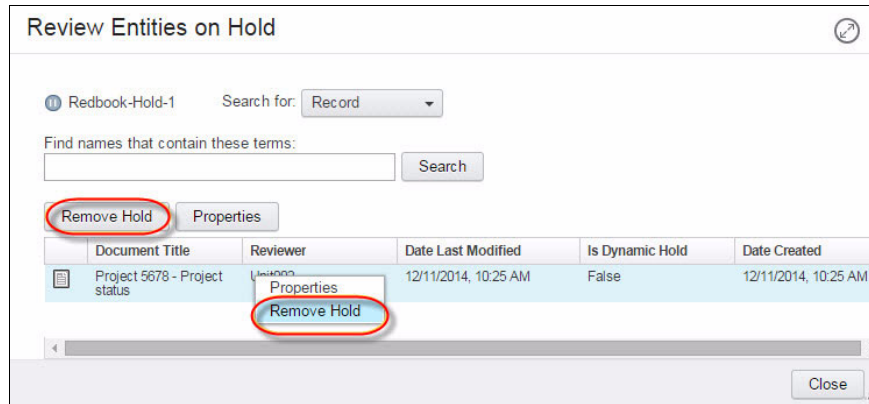


Figure 8-9 Remove entities on hold from the hold object

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.

To initiate removing a hold request, follow these steps:

1. Select the **Configuration View** icon in the launch bar.
2. Select **Holds** from the drop-down menu.

3. From the Hold list, right-click the hold object from which you want to remove the on-hold entities, and select the **Initiate Remove Hold** option, as shown in Figure 8-10 on page 225.

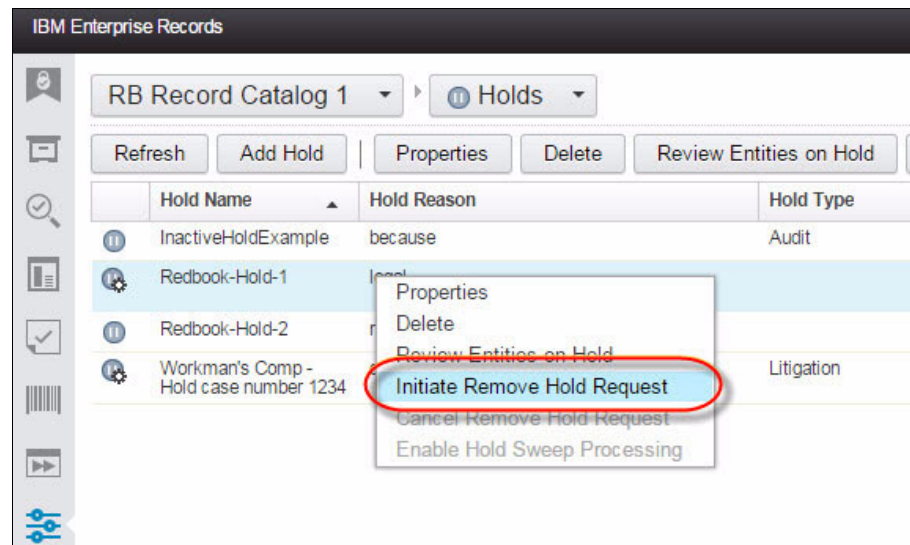


Figure 8-10 Initiating the Remove Hold Request for a dynamic hold

The next time that Hold Sweep runs, all entities that have been associated with this hold will have this hold condition removed. This action includes entities that have this hold placed on them manually.

8.2.7 Running 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.

Hold Sweep can be run in either of two ways:

- ▶ As a command-line tool that can be launched from the operating system shell
- ▶ From the IBM Enterprise Record Task Manager

Running Hold Sweep from a command line

Before you can run Hold Sweep from a command line, you must configure it to specify which hold in which FPOS it is to run. To configure Hold Sweep, complete the following tasks:

1. From a command prompt on the machine where you installed Hold Sweep, navigate to the RecordsManagerSweep folder. Enter one of the following commands:

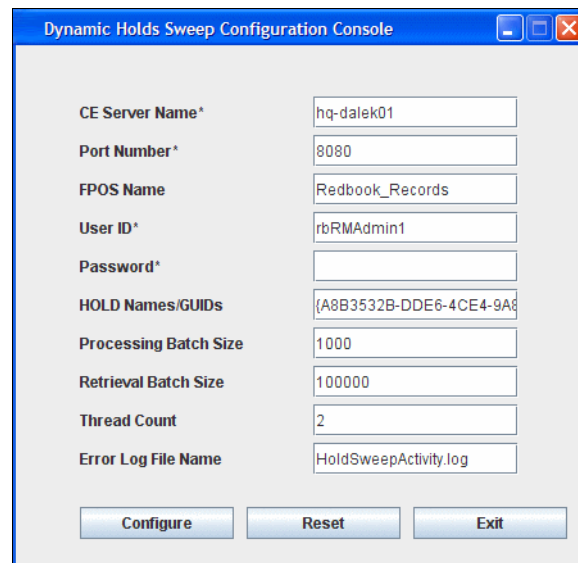
- For a Microsoft Windows operating system:

```
RecordsManagerSweep.bat -HoldSweep -configure
```

- For a UNIX operating system:

```
./RecordsManagerSweep.sh -HoldSweep -configure
```

This opens the Dynamic Hold Sweep Configuration Console dialog window shown in Figure 8-11.



The image shows a Windows-style dialog box titled "Dynamic Holds Sweep Configuration Console". It contains several labeled text input fields and three buttons at the bottom. The fields and their values are: "CE Server Name*" with "hq-dalek01", "Port Number*" with "8080", "FPOS Name" with "Redbook_Records", "User ID*" with "rbRMAAdmin1", "Password*" which is empty, "HOLD Names/GUIDs" with "{A8B3532B-DDE6-4CE4-9A8", "Processing Batch Size" with "1000", "Retrieval Batch Size" with "100000", "Thread Count" with "2", and "Error Log File Name" with "HoldSweepActivity.log". The buttons are labeled "Configure", "Reset", and "Exit".

Field	Value
CE Server Name*	hq-dalek01
Port Number*	8080
FPOS Name	Redbook_Records
User ID*	rbRMAAdmin1
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 8-11 Dynamic Holds Sweep Configuration Console

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 is for the name or IP address of the Content Engine server.

- *Port Number.* This field provides the Web Services Interoperability (WSI) port number that is used by your IBM Content Engine server. For example, the default port number for Content Engine running under IBM WebSphere® Application Server is 9080, for WebLogic is 7001, and for JBoss is 8080.
- *FPOS Name.* This 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 that are associated with the specified Content Engine server. If the name of the object store contains extended characters, use the GUID rather than the name.

GUID is the Globally Unique Identifier 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 of 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.

The execution of Hold Sweep depends on your requirements. You run it when you need to automatically add or remove holds. To avoid an impact on system performance, always run Hold Sweep *when your system use 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 affect normal business operations. When you need to stop 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.

Running Hold Sweep from Enterprise Content Manager

Enterprise Records now allows users to run Hold Sweep from the IBM Enterprise Content Manager Task Manager. This is a convenient alternative to running the tool from the command line.

Because Enterprise Content Manager calls the tool directly, it is necessary to install and configure the Hold Sweep tool in each instance of Task Manager.

Task Manager requires the Hold Sweep tool to use the EJB transport protocol (by default, hold sweep uses WSI). To set the Hold Sweep tool for EJB, complete the following steps:

1. Navigate to the RecordsManagerSweep folder, and edit the RecordsManagerSweep.bat file (or RecordsManagerSweep.sh for UNIX).
 - a. Update the following line:
Set CONNECTION_TYPE=WSI to *CONNECTION_TYPE=EJB*
 - b. Update the APP_SERVER variable:
Set APP_SERVER=WebSphere to the correct application server.
 - c. Go to the section that corresponds to your application server in the command file, and follow the instructions provided in the command file for that application server.
2. Because Task Manager provides all of the required parameters at run time, configuring Hold Sweep is not required. However, we suggest that you test the validity of all of the changes made to the command file by following the configuration instructions in “Running Hold Sweep from a command line” on page 226.

For more information, see “Configuring sweeps” in the Enterprise Records topic in IBM Knowledge Center:

<http://ibm.co/1E4kB6q>

3. After Hold Sweep is configured, you can launch the tool from Task Manager with the Enterprise Records web application. Click the **Task View** icon from the launch bar, and select **Schedule Hold Sweep** in the Schedule drop-down menu, as show in Figure 8-12.

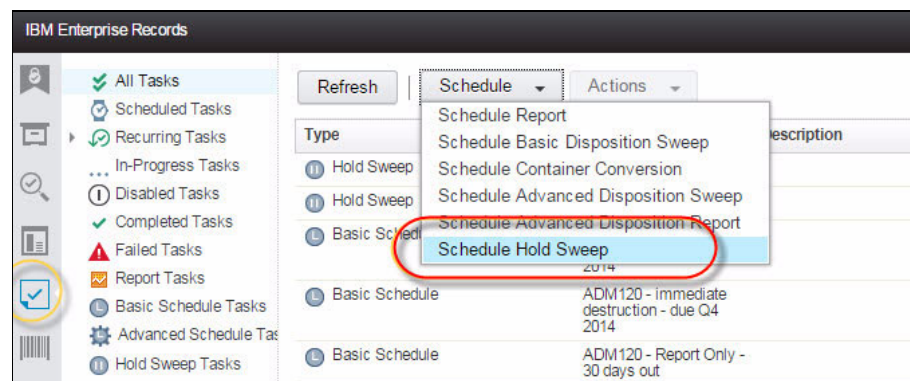


Figure 8-12 Schedule Hold Sweep

4. Provide the necessary information, as shown in Figure 8-13 on page 230:
 - *File plan repository*. File plan repository that Hold Sweep is to scan.
 - *Logs output directory*. When Hold Sweep runs, it generates an activity log. That log is automatically uploaded in a Content Process Engine repository and provided to the user for review. These parameters define the location in Content Process Engine where the log will be stored.
 - *Holds*. Select one or more holds.

▼ Set Parameters for Hold Sweep

* File plan repository: ?	RB Record Catalog 1 ▼
* Logs output directory: ?	\\RB Record Catalog 1\\Logs ▼
* Holds: ?	Redbook-Hold-1 × Select...

Figure 8-13 Set parameters for a Hold Sweep task

5. Click **Next** to provide schedule information, as shown in Figure 8-14. Provide the name of the task. If you want to run that tasks regularly, select **Run on schedule** and provide the repeat information.

Provide the user name and password that the Hold Sweep will use to search for the records to be held. This user must be able to update the properties of all the records that will be found.

Set Parameters for Hold Sweep

Set Schedule

Schedule Information

* Name: Run hold sweep for Redbook hold 1

Description:

☒ Run once

* Start time:

☒ Start immediately

☐ Run on a schedule

* Repeats: Daily

* Start date:

End date:

Login Information

* User name MyUsername

* Password

Figure 8-14 Configure a Hold Sweep schedule

- Click **Schedule Sweep** to complete initiating the task.

8.2.8 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 and containers filed below that point in the file plan hierarchy inherit the hold.

8.2.9 Disposal trigger aggregation level effect on holds in advanced dispositions

A factor that influences hold functions 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 disposition. The problem is that an organization might have more records on hold than they are really aware of, resulting in records that they thought were eligible for disposition but not being processed. To avoid this behavior, you have two options:

- ▶ Evaluate the impact of aggregation at the record level. This is not recommended because of the performance impact of this type of aggregation.
- ▶ Check **Destroy containees not on hold**, which is available on a Destroy or Auto Destroy type of action. When this option is selected, the destroy workflow or the automatic destroy sweep destroys any records that are not filed as *On Hold* in the container. The container itself will not be disposed of, because it still contains the records that were put on hold.

8.3 Performance considerations

From a database perspective, Hold Sweep is a resource-intensive operation. For this reason, it is best to schedule Hold Sweep to run when system use is low. To set the correct expectation, if you have a conditional hold that might put approximately a million records on hold, 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. Change them only 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 effect 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. Therefore, it is important to ensure that these searches run optimally, which means that the best practice is to routinely tune the underlying database.

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. Therefore, it is critical for large systems to build a cluster index on GUID to improve performance.

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 separation, such as records, folders, and categories, there are three independent searches, which Hold Sweep performs sequentially. For example: A hold is being created for a litigation action that is underway against a contractor to Fictional Insurance Company XYZ. 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.

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 places on holds all records where *XYZVendorID=A123456*, and it then searches for and places on holds 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.



Audit requirements

This chapter introduces the auditing requirements within a records management system. We describe the extensions to the built-in auditing functions provided by IBM Enterprise Records and explain how this audit information can be accessed and reported.

We cover the following topics:

- ▶ Introduction to audits
- ▶ Audits of an IBM Enterprise Records system
- ▶ Accessing the audit log
- ▶ Pruning the audit log

9.1 Introduction to audits

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 audits are required:

- ▶ To ensure compliance with the organization's standards
- ▶ To ensure that records will be accepted as evidence in a court of law

9.1.1 Compliance audits

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 recordkeeping are of value even if the need to produce electronic records in court never arises. The effort and resources required to comply quickly bring business benefits, whether the organization is in court or not.

9.1.2 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 functioning or malfunction, the evidential weight or value put on that document can be lost or at 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.

9.2 Audits of an IBM Enterprise Records system

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. The IBM Content Platform Engine architecture provides this framework, which is augmented when Enterprise Records 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 Content Platform Engine. Instead, it describes auditing for Enterprise Records systems.

When a Records Manager data model is imported into an object store, the standard Content Platform Engine 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** → **IBM Enterprise Records** → **Auditing**, you'll find details about the system events that must be enabled normally on the relevant object classes within the file plan object store (FPOS).

Where electronic records are being managed, consideration must be given to the auditing requirements for the records that are contained within the FPOS. By default, auditing is not enabled for an object store. Therefore, to capture audit events (whether system or RMAudit), auditing must be enabled on the object store.

For details about how to enable auditing on an object store, select **ecm_help** → **FileNet P8 Administration** → **Content Engine Administration** → **Auditing**.

Note: A preferable practice for auditing is to enable only 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 have the child objects inherit that setting. This action would cause an excessive amount of audit data to be generated, which would severely affect the performance when running audit reports. Also, the data that is eventually returned to the user would contain superfluous information that the user must read through to get to the pertinent information.

9.3 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

You can access these entity entries one at a time through the Content Platform Engine administrative console.

9.3.1 Accessing audit information from IBM Enterprise Records

To access an entity's audit entries in Enterprise Records systems, complete the following steps:

1. Browse through the appropriate container to find the object (record, category, folder, or volume) to query, or go to the Search page to find this object.
2. After you have found the record, go to the **Properties** page and select the **History** tab. Provide search criteria if necessary, and then click **Search** (see Figure 9-1 on page 239).

Properties	Details	Disposition	Vital Record	Security	Holds	History
Search Criteria Filter: Action by All Show dates from: to Search Reset						
Action	Date or Time	Initiator	Status			
Creation Event	10/2/2014, 1:26 PM	RMAAdmin@IER01dom.it	Success			
Update Event	10/2/2014, 1:28 PM	RMAAdmin@IER01dom.it	Success			
Update Event	10/2/2014, 1:29 PM	RMAAdmin@IER01dom.it	Success			

Figure 9-1 Searching audit history for an entity

9.3.2 Accessing audit information with the Content Platform Engine

A second method of accessing audit information is through the Query Builder in Enterprise Manager or the Content Platform Engine administrative console. The 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. It also allows an administrator to either query for the allowed audit events or to specify a type of event to search. The following types of events can be searched:

- Creation
- Update
- Delete
- RMAudit

To search for all audit events within the Query Builder by using the Content Platform Engine administrative console, complete the following steps:

1. Log in to the administrative console for Content Platform Engine. Select the object store to query from, and start a new **Object Store Search**. In the **Class** drop-down menu, select the type of event that you want to query or select **Event** to search for any event. In Figure 9-2 on page 240, we want to search only for “Update Event.”

Run Save Save As Actions Close

Search: New Object Store Search

Simple View SQL View Bulk Actions (Disabled)

Construct or edit a query step by step by entering search criteria. You can optionally switch to the SQL View tab after you

Class: ?

Criteria ?

Update Event

Update Event

Update Security Event

Column	Condition	Value
A <none>	<none>	
B <none>	<none>	
C <none>	<none>	
D <none>	<none>	

Criteria Grouping ?

Figure 9-2 Start building an event search in ACCE

2. Add criteria. Use the Criteria section of the search Query Builder to specify extra criteria. Figure 9-3 shows an example of searching for a specific category (the Source Object of the event) by its GUID.

Run Save Save As Actions Close

Search: New Object Store Search

Simple View SQL View Bulk Actions (Disabled)

Construct or edit a query step by step by entering search criteria. You can optionally switch to the SQL View tab after you begin query construction here.

Class: ?

Update Event

Criteria ?

Column	Condition	Value
A Source Object Id	Equal To	{A7A601D8-FF4C-4544-AF04-4D183C987875}
B <none>	<none>	
C <none>	<none>	
D <none>	<none>	

Criteria Grouping ?

Figure 9-3 Specify criteria in the search builder

3. Click **Run** to launch the query. The result is a list of all of the events that occurred, based on the specified criteria, as shown in Figure 9-4.

RunSaveSave AsActionsClose

Search: New Object Store Search

Simple ViewSQL ViewBulk Actions (Disabled)Search Results x

Actions

SQL statement: SELECT TOP 500 [This], [CmAuditSequence], [Creator], [DateCreated], [DateLastModified], [EventStatus], [Id], [InitiatingUser], [LastModifier], [Name], [Owner], [UpdateEvent] FROM [UpdateEvent] WHERE [SourceObjectId] = {A7A601D8-FF4C-4544-AF04-4D183C987875} OPTIONS(TIMELIMIT 180)

	ID	Audit Sequence	Creator	Date Created
	(1892DB0D-7844-46FC-BEC6-068D4D11171D)	3	RMAdmin	October 2, 2014 at 1:28:30 PM Pacific Daylight Time
	(273099FE-DB92-4540-8D34-7734F4922679)	4	RMAdmin	October 2, 2014 at 1:29:48 PM Pacific Daylight Time

Figure 9-4 See the search result

4. Click one of the events to see details about what properties were updated.

9.4 Reporting by using the audit data

The audit data can be used in reports. For example, the “Actions Performed by a User” report that is included in the software uses Content Platform Engine auditing to browse all of the actions that users have performed in the system.

9.5 Pruning the audit log

Because the audit log keeps growing, a new feature that was introduced with Content Platform Engine 5.2 allows pruning log entries to keep the log file to a reasonable size. This enables a Content Platform Engine administrator to configure an audit disposition policy and run an audit disposition sweep.

For more information, see “Pruning audit entries” in IBM Knowledge Center:

<http://ibm.co/1e1MXt1>



Reporting

This chapter describes the reporting feature of IBM Enterprise Records. It covers the following topics:

- ▶ Reporting capabilities and considerations
- ▶ Running IBM Cognos reports
- ▶ Running reports from Crystal Reports

10.1 Reporting capabilities and considerations

IBM Enterprise Records provides predefined reports that provide a statistical view of activities performed while using the application. 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.

IBM Enterprise Records now supports two report engines:

- ▶ IBM Cognos
- ▶ Crystal Report

Depending on the volume of data (records and containers) of the Enterprise Records 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 of 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 titled “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.

10.2 Running IBM Cognos reports

IBM Cognos reports are based on the report engine framework, which exports the data from the IBM Content Platform Engine to a report database. The Cognos report templates (or any reporting tool) can then read that intermediate database and format it for the user.

Running a report from the Enterprise Records application links those steps together to provide an end-to-end solution from the request to the display of the report.

For more information about this architecture, see *IBM Enterprise Records Reporting Framework Developer's Guide* in the Enterprise Records installation media (/CognosReports/IER Reporting Framework Developers Guide.pdf).

10.2.1 Configuration

Enterprise Records must be configured to run Cognos reports. Starting from the desktop Configuration window, complete the following steps:

1. Select **Open Administration View**.
2. Expand the **Desktop** node, and select the desktop from which you want to run the reports.
3. Select the **Reports** tab.
4. For “Report server type,” select **Cognos** in the drop-down list and specify the **Cognos Server Configuration** and **Report Engine Configuration** options. For details about those settings, see “IBM Enterprise Records installation and update” in IBM Knowledge Center:
<http://ibm.co/1JeEj6s>
5. **Save** your settings.

Note: The Enterprise Records server must be configured before running any report. See the IBM Knowledge Center page cited previously for how to set up the application for Cognos.

10.2.2 Predefined reports

Enterprise Records provides six predefined Cognos report templates, as shown in Figure 10-1 on page 246.

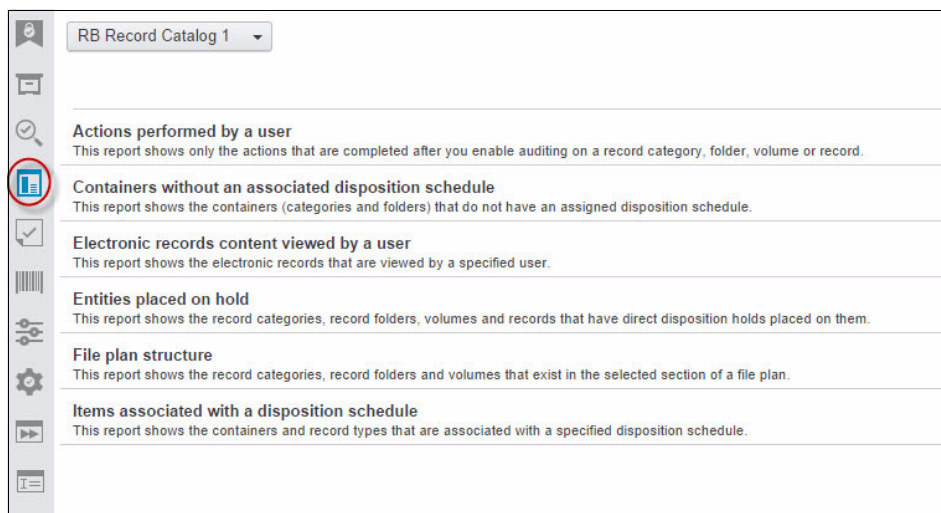


Figure 10-1 Predefined Cognos reports

10.2.3 Running reports from IBM Cognos

IBM Cognos reports can be run interactively or as a batch background task by using the Task Manager.

Running a Cognos report interactively

To run a Cognos report interactively, follow these steps:

1. Click **Report**, and then select the report that you want to run.
2. Provide the required parameters for that report.
3. Generate the report.

Note: Rather than running the report interactively, you have the option to use the Task Manager feature to run it in the background. See the next section.

Running a Cognos report as a background task

Cognos reports can be generated as a background process using Task Manager. You have the option to schedule the report to run later (at night or during a weekend, for example), when the charge on the system is lower.

When running the reports through Task Manager, the report is generated as a PDF file that is saved to the Content Platform Engine where you specify. This file can then be downloaded to preview or print.

To schedule a report task, complete the following steps:

1. Open **Task View**.
2. Click the **Schedule** tab and, from the drop-down menu, select the **Schedule Report** option, as shown in Figure 3.

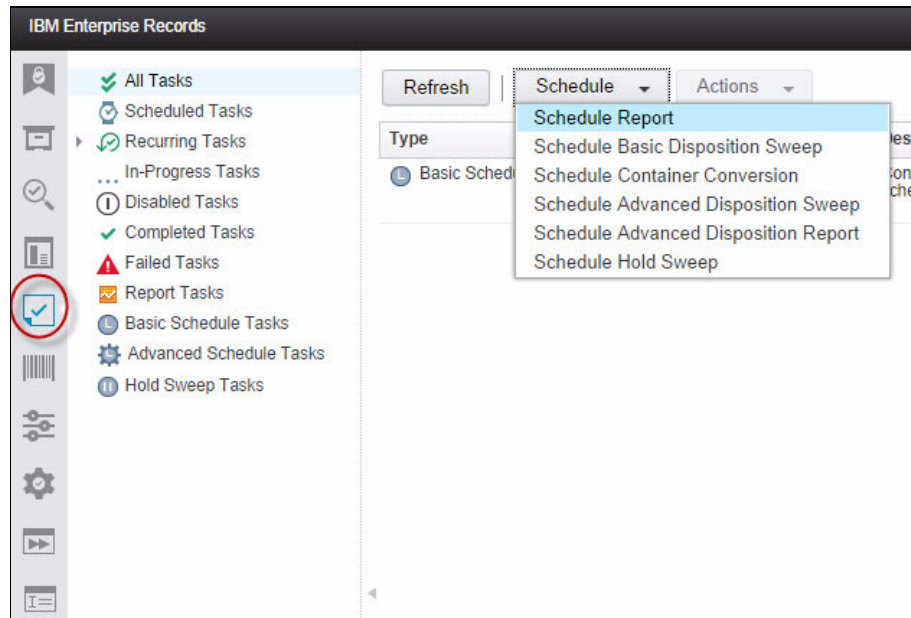


Figure 10-2 Schedule report options

3. Create a scheduled report task.
4. Select the report to run and provide the values required by the selected report.
5. In the next window, provide the name of the task to create. If you want to run the report later, clear the **Start Immediately** check box to set the start time and enter the user name to use at the report run time.
6. In the next window, specify where you want to save the report PDF file in the Content Process Engine. Provide a name for it, and if it contains confidential information, set up the security for the report.

Note: Because the report PDF file is a document, it cannot be filed under the Records Management folder of the default file plan object store. You must select a different object store or create a root folder in the file plan object store (FPOS) where the reports will be saved.

7. Then, click **Schedule Report**.

When the task is complete, you can access the report PDF file by either of two methods:

- ▶ By searching for the report in the file location that was defined while creating the task
- ▶ By selecting the task in Task Manager and clicking the **Result** tab, selecting the report, and then clicking **Download Report**, as shown in Figure 10-3.

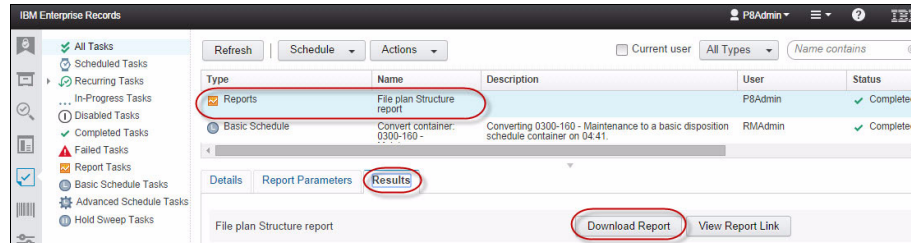


Figure 10-3 Download a Task Manager report

10.2.4 Adding a new Cognos report

The Cognos reports can also be customized to an organization's requirements or, if necessary, new ones can be created. For the details about how to customize or create your own reports, see *IBM Enterprise Records Reporting Framework Developers Guide* in the IBM Enterprise Records installation media (/CognosReports/IER Reporting Framework Developers Guide.pdf).

10.3 Running reports from Crystal Reports

The interface that Crystal Reports uses to access the data within the IBM Enterprise Records databases is a standard Java Database Connectivity (JDBC) interface.

10.3.1 Configuration

To run reports from Crystal Reports, IBM Enterprise Records must be configured accordingly. From the desktop Configuration window, complete the following steps to configure IBM Enterprise Records for Crystal Report:

1. Select **Open Administration View**.
2. Expand the **Desktop** node and select the desktop from which you want to run the reports.
3. Select the **Reports** tab.

4. For “Report server type,” select **Crystal Reports** in the drop-down menu, and specify the Crystal Reports server name.
5. Save your settings.

Note: The IBM Enterprise Records server and Crystal Reports server must be configured before you run any report. To set up the application for Crystal Reports, see “IBM Enterprise Records installation and update” in IBM Knowledge Center:

<http://ibm.co/1JeEj6s>

10.3.2 Predefined reports

IBM Enterprise Records provides 17 predefined Crystal Reports templates that can be accessed through the Reports feature within Enterprise Records, as shown in Figure 10-4 on page 250.

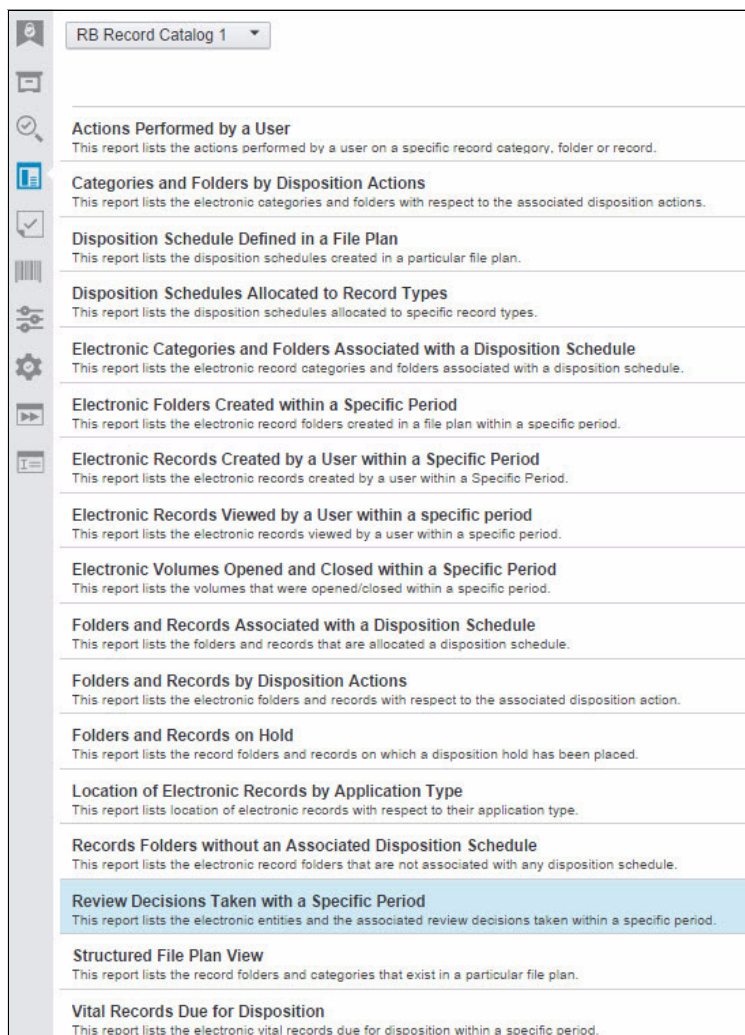


Figure 10-4 Predefined Crystal Reports report templates

To run a Crystal Reports report, follow these steps:

1. Select the report from the list of predefined reports.
2. Provide the values required by the selected report.
3. Generate the report, as shown in Figure 10-5 on page 251.

Figure 10-5 Report sample: File plan structure

Note: Different from the Cognos reports, you cannot schedule Crystal Reports reports. Running them interactively from the **Report** feature is the only option.

<http://ibm.co/1bPVKwP>



Physical records

This chapter provides an overview of working with physical records in IBM Enterprise Records. It describes the key features that are available to use in physical records management and explains how to use those features in your environment.

This chapter covers the following topics:

- ▶ Overview of physical records management
- ▶ Enterprise Records physical records capabilities
- ▶ Tracking physical records

11.1 Overview of physical records management

Even in this electronic era, most organizations still have physical items that need to be managed as records. IBM Enterprise Records provides additional features, beyond pure electronic records management, for the management of physical records within a single application.

Physical records are those artifacts that capture or represent a business decision or decisions, which exist in a non-digital format, or where there is value or significance associated with the non-digital format. Some examples of physical items you will be familiar with include articles of incorporation, certificates of title, photographic plates, original audio tapes, crime scene evidence, historical books and artifacts, and so on.

Figure 11-1 provides a simple overview of some basic management requirements for physical records.

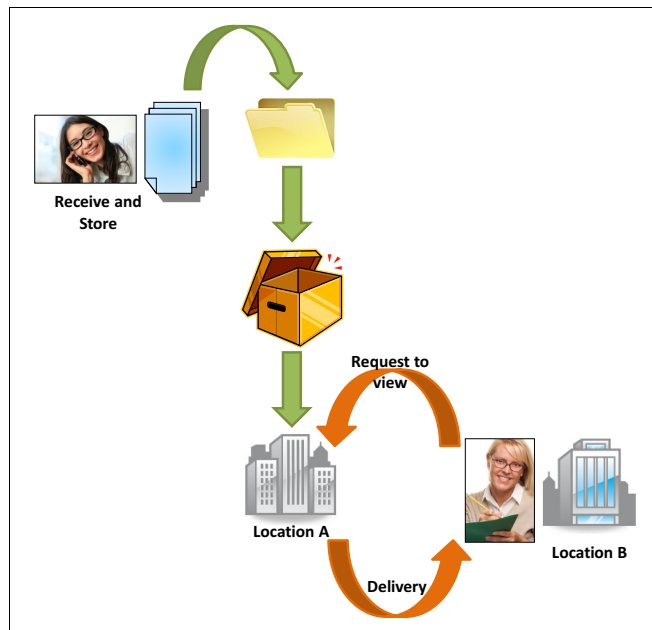


Figure 11-1 Basic management requirements for working with physical records

A user receives or identifies a physical record which should be managed as an organizational record. The user adds the new physical record to the system and nominates where the item should be filed. Eventually, the item's file might be placed into a box and the box might be moved to a storage location. If another user in the same or different location wants to retrieve the item, they submit a request for the item. On approval of the request, the item is delivered to the requester. After the requester is done with the item, they return it to the storage location.

This simple scenario highlights that, aside from the obvious format differences between electronic and physical records, the management requirements of physical records includes the following attributes, at a minimum:

- ▶ Physical records have a home location, which is where they are normally stored.
- ▶ Physical records might be moved to another location for a period of time, so they must support recording of a temporary location.
- ▶ Users might request access to physical records.
- ▶ Physical records are typically stored or managed in containers (this might not have relevance for electronic records).
- ▶ To aid movement requests, physical items might use bar code labels.

Note: To work with physical items in Enterprise Records, users are required to have access to an Enterprise Records desktop.

11.2 Enterprise Records physical records capabilities

Enterprise Records includes features to support the requirements of managing physical records in the same application as you manage your electronic records. For management of physical records, the application includes the following feature constructs:

- ▶ Boxes
- ▶ Physical and hybrid folders and folder volumes
- ▶ Physical records
- ▶ Bar code support
- ▶ Workflow to support requests for records
 - Reservations
 - Charge outs

Each of these items is described further in this section.

11.2.1 Containers: Boxes

When working with physical items, it is common to see people filing paper into folders and storing the folders on shelves or in file drawers. In situations with large collections, the ongoing management of the physical items might be outsourced to external parties, and you might see folders placed in boxes that are then shipped to the external party.

Enterprise Records models a container construct for physical records and provides a mechanism to model physical entities that contain other physical entities. Although more material is handled digital today, many organizations still have requirements to manage physical records. This can be managed in Enterprise Records.

These physical items persist for two main reasons:

- ▶ It is not practical to store the item on a shelf in a record folder for any of these reasons:
 - Due to the type of record, such as audio or video cassettes or photographic plates
 - Because original items that have historical significance, such as original company logo or advertising materials, might have different formats and sizes
 - Because the material is bulky, such as crime scene evidence
- ▶ It is organizational policy to add the physical records to a folder and then store the folders in boxes, before moving them to off-site storage.

In Enterprise Records, unlike other types of record folders, a box can contain other folders. These entities are physical folders and boxes, but not electronic or hybrid folders and records. Boxes do not use volumes.

There are two methods for creation of new file plan containers in Enterprise Records, both of which support the creation of boxes, hybrid record folders, and physical record folders:

- ▶ In the first method, a user submits a *workflow request* for a new box to be created.
- ▶ The second method relies on the user having permission to create the container in the Enterprise Records *desktop*.

Workflow request for creation of a new box

The request for a new box to be created is initiated by a user working in the Content Navigator desktop.

1. Initiate the Create Folder request by selecting the **Request Record Folder** option from the drop-down menu on the toolbar, as shown in Figure 11-2.

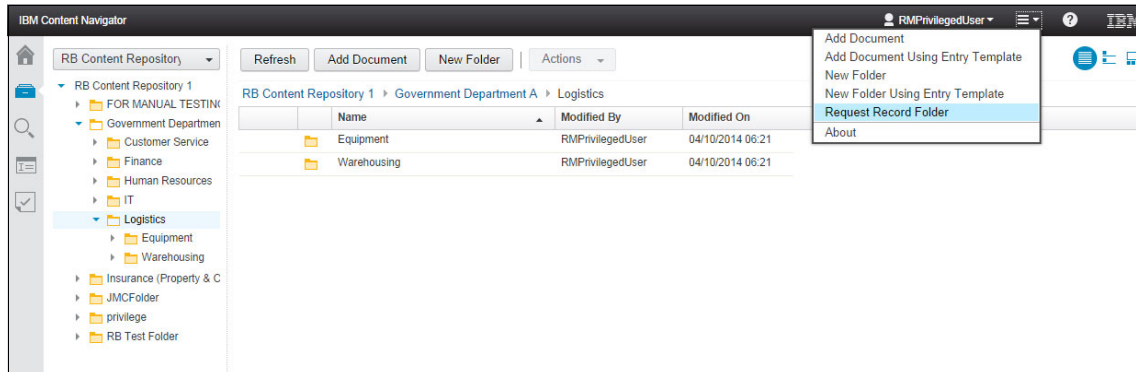


Figure 11-2 Request Record Folder

2. After you select this option, complete the details of the request in the next window, as shown in Figure 11-3 on page 258, where you provide the name of the box that you want to create, where the box should be in the file plan, and stipulate to identify the item as a *box*. Figure 11-3 on page 258 illustrates that the Physical Container item was expanded to show the Box option, which is highlighted.

In completing the required fields on this submission interface, all of the data fields are preceded by a red asterisk, which indicates all fields are mandatory and require completion before the Launch Workflow button is activated. Also, you can select only items that are not shown as inactive.

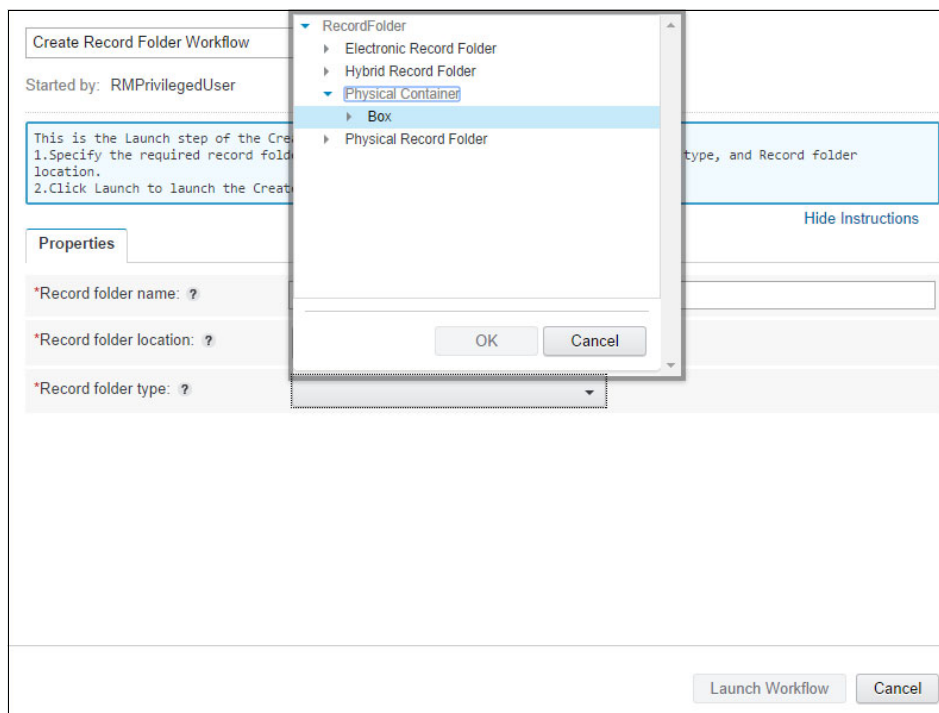


Figure 11-3 Details required to submit a Request Record Folder creation submission

3. Select **Box** then **OK**. If all of the required fields are completed, the Launch Workflow button is activated, allowing you to submit the request. After the request for a Box was submitted, the Records Administrator reviews the request and, if approved, creates the Box container in the specified location of the File Plan.

The Records Administrator then notifies you, as the requester, through the workflow.

Manual creation of a box, using the desktop

To manually create a box in Enterprise Records, you need access to an Enterprise Records desktop. In our environment, we have created a desktop for the Records Management Privileged User (RMPrivilegedUser), which provides a desktop interface for our privileged user (potentially, the department Information Coordinator).

1. To create a new box, access the file plan, navigate to the category where you want to create the box (**0300-130 Installation** in Figure 11-4 on page 259), and select **Add Record Folder**.

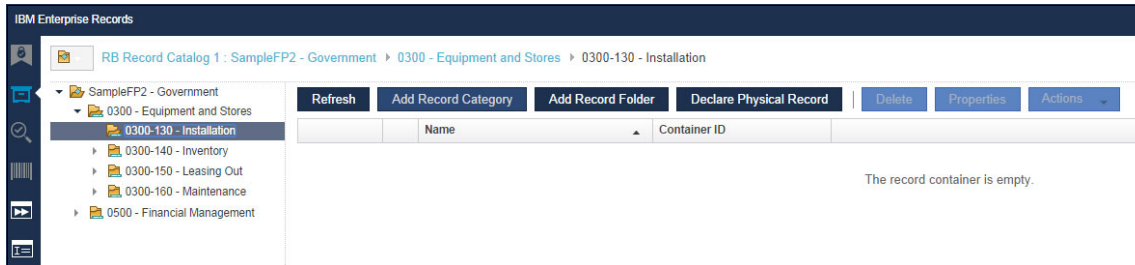


Figure 11-4 Initiate manual creation of a new Box

2. In the manual box creation window, shown in Figure 11-5 on page 260, you must complete required information in all fields that are preceded by an asterisk to complete the task.

To start with, select the **Class** drop-down menu, expand the **Physical Container** element, and then select the **Box** element. After you click the Box element, the OK button becomes active, and you can click **OK** to complete the other details.

One of the items that you must complete for the box to be created is the *Home Location* field. Choose **Select** next to the Home Location field to see the list of available locations. The Home Location is where the box will be stored.

General

* Save in: 0300-130 - Installation

Properties

* Class: Box

* Record: 0300-130-100 Installation New Plant - Costa Mesa

* Folder: 0300-130-100

Description: Records documenting the installation and initial configuration of equipment and plant.

Date Of: 04/10/2014 08:09

* Review: RMPrivilegedUser

* Home: Location A

Disposition

Basic disposition: No schedule

Advanced disposition schedule: Inherited

* Disposition instructions: 0300-130 - Installed + 2Y

Disposition authority: AFDA1149

Vital Record

Security

Figure 11-5 Manual box creation window

The details of the disposal schedule in Figure 11-5 were inherited from the preceding container in the file plan. In this example, our RMPrivilegedUser does not have authorization to modify the disposal schedule, so the Disposal Schedule button is inactive.

- When all of the details are correct, you can select **Add** to complete the creation of the box. Notice that the part of the screen captured in Figure 11-5 does not display the Add button, which is in the lower-right of the window.

Note: The preferred practice is to provide only privileged users with modification access to those file plan nodes for which they require such access, and that these users be familiar with the organization's recommended naming conventions for the file plan.

11.2.2 Containers: Physical and hybrid folders and folder volumes

Enterprise Records supports physical and hybrid folder types for management of physical folders. Within Enterprise Records, 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 the physical storage structure of your organization records, you can model the hierarchy of physical folders in Enterprise Records.

Within Enterprise Records, a hybrid folder is similar to an electronic folder but it can contain both electronic and physical records, and contains one or more volumes. 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 Home Location.

Figure 11-6 shows the various icons representing physical containers in Enterprise Records. The first icon represents a hybrid folder, the second a box, and the third icon represents a physical folder.



Figure 11-6 IBM Enterprise Records physical container icons

In Enterprise Records, when you create a record folder, the volume is automatically created for you, and when you add a new volume, any existing open volume is closed. Record volumes inherit their values from their parent folders. Boxes do not contain volumes.

Enterprise Records supports the movement of all of these container types in charge out requests. This is covered in “Charge outs and reservations” on page 275.

Physical record folder creation

The creation of a physical record folder or a hybrid record folder in Enterprise Records is similar to the creation of a Box, and accessed through the same interfaces. You can request the creation of a physical record folder or hybrid record folder from within the IBM Content Navigator records-enabled desktop or from within an Enterprise Records desktop, as described in 11.2.1, “Containers: Boxes” on page 256.

Figure 11-7 shows the window for adding a physical record folder. The fields prefaced by an asterisk are mandatory fields. Notice in the figure that the physical record folder being created is added to a box (0300-130-100 - Installation New Plant - Costa Mesa), and it uses the ready-for-use Physical Record Folder class. This metadata, defined at the record folder level, can be extended by creating a new folder class based on this class (subclass) and adding the additional metadata values that are required by your organization.

General

* Save in:

0300-130-100 - Installation New Pla

Properties

* Class:

Physical Record Folder

* Record Folder Name:

Plant Electrical Plans - Lighting

* Folder Unique Identifier:

0300-130-100 Electrical Plans

Description:

Records documenting the installation and initial configuration of equipment and plant.

Date Opened:

07/10/2014 05:26

* Reviewer:

RMPrivilegedUser

* Home Location:

Location A

Select...

Barcode:

New Plant Costa Mesa

Location:

Location A

Select...

Disposition

Basic disposition schedules are high-performance schedules that are easy to use. Advanced disposition schedules offer more capabilities and flexibility, but they are more complex and negatively impact performance

No schedule

Advanced disposition schedule

* Disposition instructions:

Inherited

0300-130 - Installed + 2Y

Browse Schedules

Disposition authority:

AFDA1149

Vital Record

Security

Figure 11-7 Physical Record Folder creation

When creating a physical folder inside a box, some of the values are inherited from the box, such as the Disposal Schedule, Home Location, and (current) Location fields.

In an electronic records management system, the requirement to partition the information contained in records management folders into volumes is not as obvious as it is in a physical records management environment. While it is possible to model the physical hierarchy of folders within Enterprise Records, any folder hierarchy modeled in the application should be modeled to meet business

262 Using IBM Enterprise Records

requirements. Our preferred practice is to model a simpler file plan to eliminate confusion in when classifying records in the file plan.

11.2.3 Bar codes

Enterprise Records provides ready-for-use support for storing the value of a bar code in fields on all of these physical item entities:

- ▶ Boxes
- ▶ Physical record folders
- ▶ Hybrid record folders
- ▶ Marker records
- ▶ Home location and current location

Figure 11-8 illustrates how all of these items can have their own bar code.

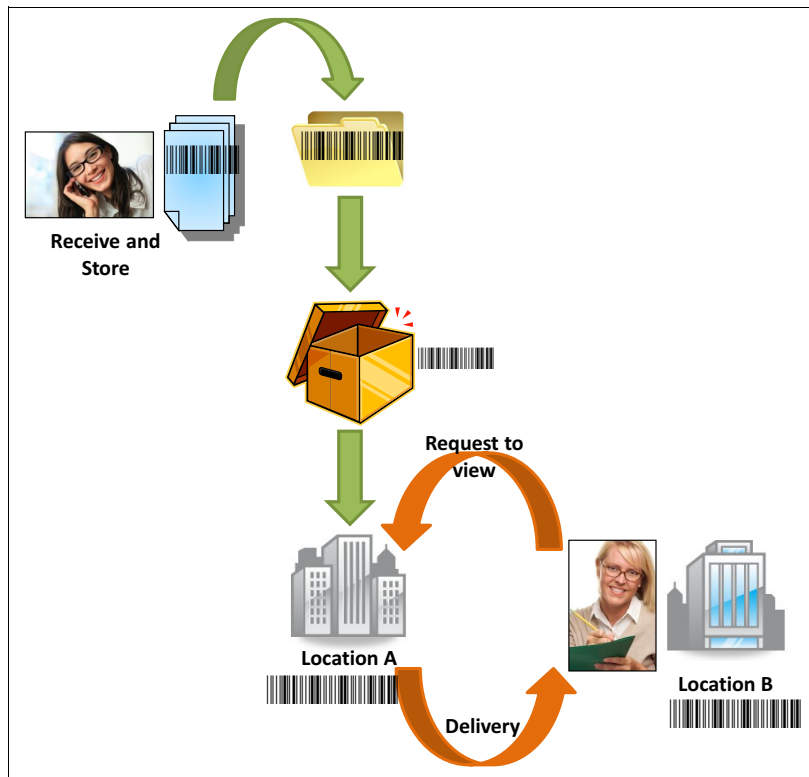


Figure 11-8 Physical items supporting bar codes

If your organization is equipped to print bar codes, by storing the value of a field in the bar code metadata, you can use a bar code print to generate labels to attach to your physical items or locations. If your organization also has suitable hardware, you can use this hardware to monitor, track, and update individual items in Enterprise Records.

Bar codes in IBM Enterprise Records

To work with bar codes in Enterprise Records, you need access to the Physical Items feature in your Enterprise Records desktop, as shown in Figure 11-9.

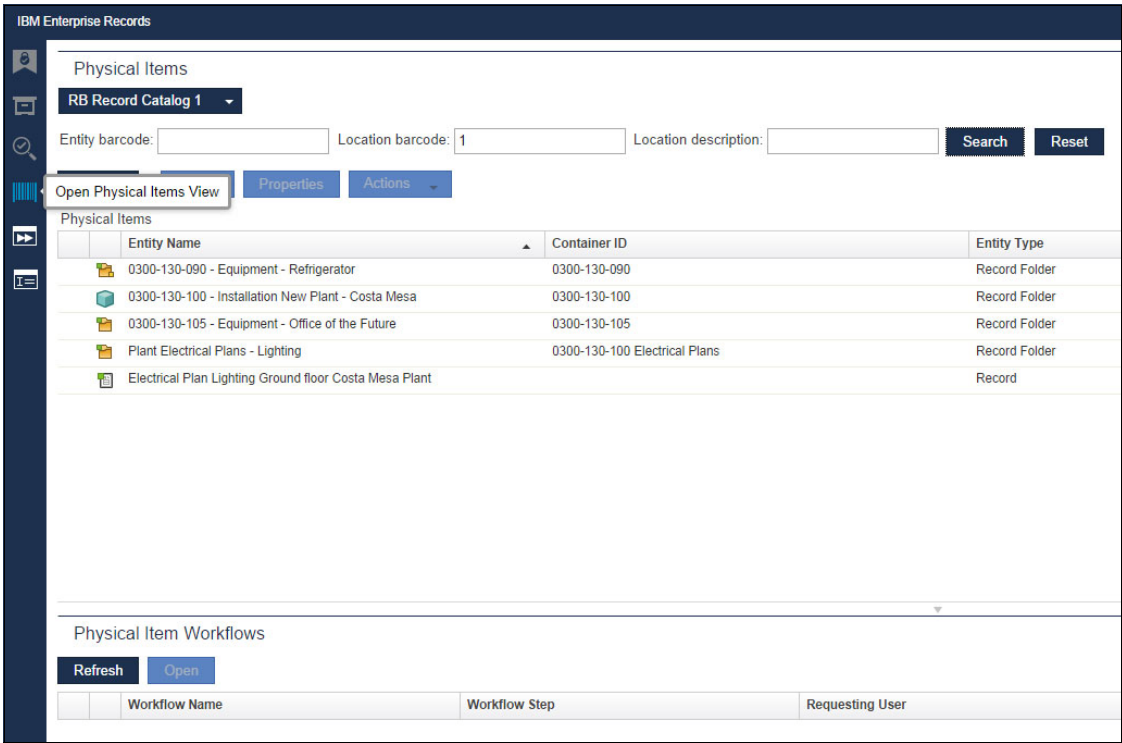


Figure 11-9 Physical Items feature in the IBM Enterprise Records desktop

When you select the **Open Physical Items View** menu, a window opens that enables you to enter values for physical items or, if you have appropriate hardware, to scan the items' bar codes. In Figure 11-9, we typed the value 1, which has retrieved all items that containing a “1” in the Location field. You can also enter values in the Entity Bar Code field or in the Location Description field.

Creating bar codes

When adding a new Marker item to Enterprise Records you have the option to add a bar code, as shown in Figure 11-10. When you enter the bar code value in this window, you type in the readable value of the bar code, which is the value you print when generating a bar code label.

* File plan location: ?	Plant Electrical Plans - Lighting
* Record class: ?	Marker Record
Document Title: ?	Electrical Plan Lighting Ground floor Costa Mesa Plant
Description: ?	Lighting technical drawing for the ground floor of the Costa Mesa plant
*Home Location: ?	Location A x
Barcode: ?	EPL GF CMP
Location: ?	Location A x

Figure 11-10 Marker item creation

In the diagram, you can see the Barcode field, but completing it is not required. You can change the attributes of the field to Required, if necessary, in the document class for marker items.

Note: Any restrictions on the field value applied to barcodes fields are bar codes dependent upon the bar code font used. To print bar codes you must have a bar code font installed, and you will need access to a bar code printer.

Enterprise Records also supports bar codes for locations, so the location items also have a Barcode field, as shown in Figure 11-11. Enter the value in the bar code field according to your organizations naming conventions.

*Location Name: ?	Baker Warehouse San Francisco
Barcode: ?	BWS 1-11
Description: ?	Baker Warehouse Hillside Blvd San Francisco CA
Reviewer: ?	RMManager

Figure 11-11 Location properties pane

The metadata shown in Figure 11-11 on page 265 is a sample of what can be captured for locations and can be extended to meet the requirements of your organization.

Note: Only users authorized to use the Enterprise Records desktop Configuration feature can create locations.

11.2.4 Searching

Searching for Physical Items in Enterprise Records uses the underlying search features of IBM Content Navigator and IBM Content Foundation applications. Physical items do not have content elements available for full-text retrieval in the system, so we focus here on metadata and keyword searching.

Some of the distinguishing features of physical items are the Home Location, Location, and bar code details that are stored as metadata for physical items. You can use one of these details when searching across broad record types, such as looking for all types of record folders, to ensure that your results are narrowed to physical items only. The example that follows shows how to create a search for a broad set of physical items and choose the properties to display to return a small meaningful search result set.

When the search results set is displayed, you can select one or more items and export the results for use in *ad hoc* reporting.

Note: To search for physical items, users must have access to an Enterprise Records user desktop.

When your system administrator creates a desktop, they can define whether to allow the desktop users to create new searches or to limit the search capabilities of the desktop users to running only predefined searches. Limiting access for defining new searches and to creating searches with predefined fields ensures that users cannot run inefficient searches over large collections. Enterprise Records presents users only with result sets that containing materials that the user has rights to view or modify.

Create a new search example

Figure 11-12 on page 267 shows a sample desktop for a Privileged users. From the Search View feature, you get access to create a new search or to re-use an existing search.

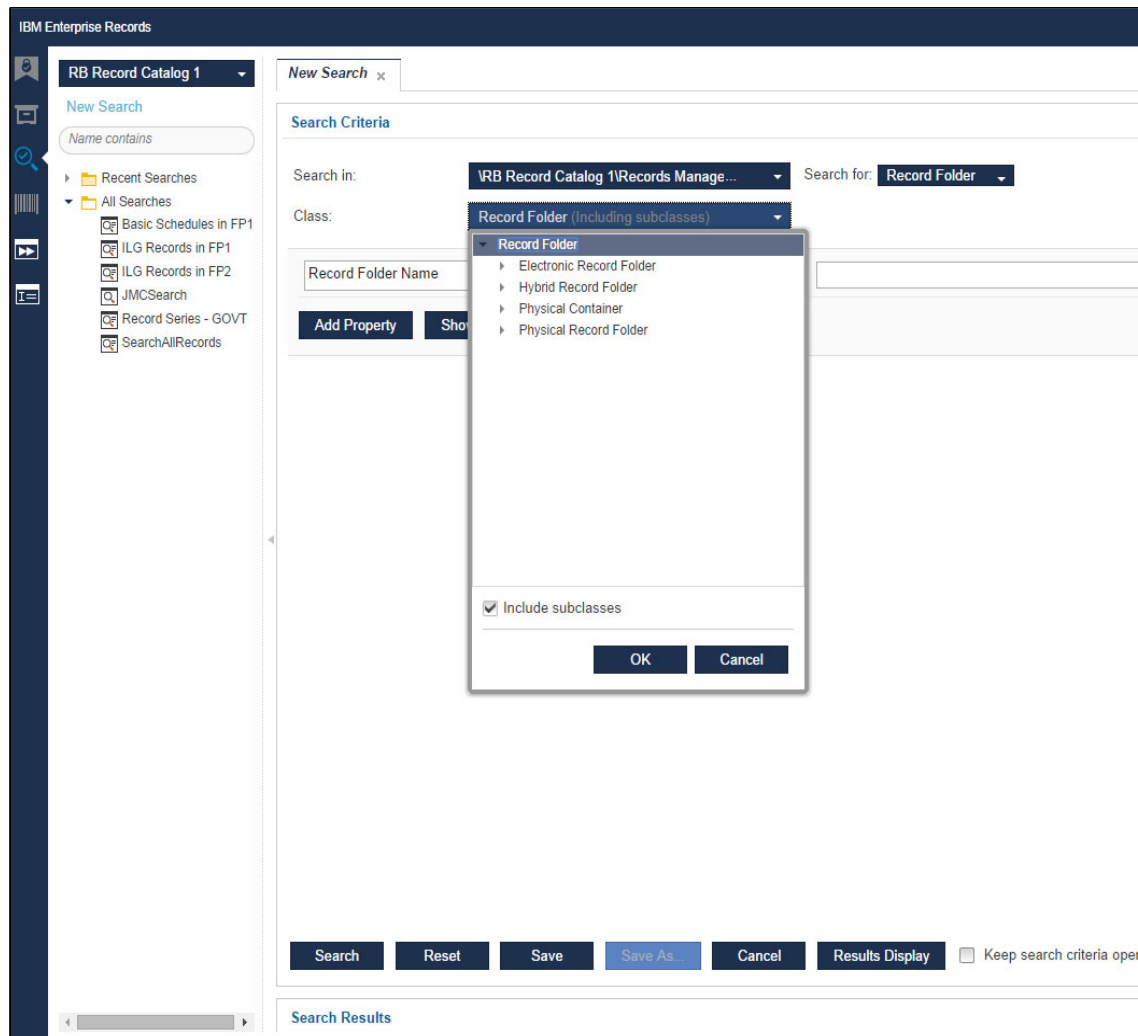


Figure 11-12 Searching for boxes, hybrid folders, or physical record folders

To search for boxes, hybrid folders, and physical folders in a particular part of a file plan, and to present back the results with disposal schedules, rules, and reviewers, you must create a new search and choose the following items:

1. The scope of your search. State how much of the FPOS or file plan you want to search.
2. The type of item that you are searching for. In this example, you can select **Record Folder**.

3. If you want to search for a particular class of item, which in this example is all record folder types, make sure that you check **Include subclasses**.
4. To refine the search and exclude electronic record folders, select the **Record Folder Name**. From the drop-down menu, click the **Location** property, as shown in Figure 11-13, and select **Is Not Empty** from the Operator drop-down menu.

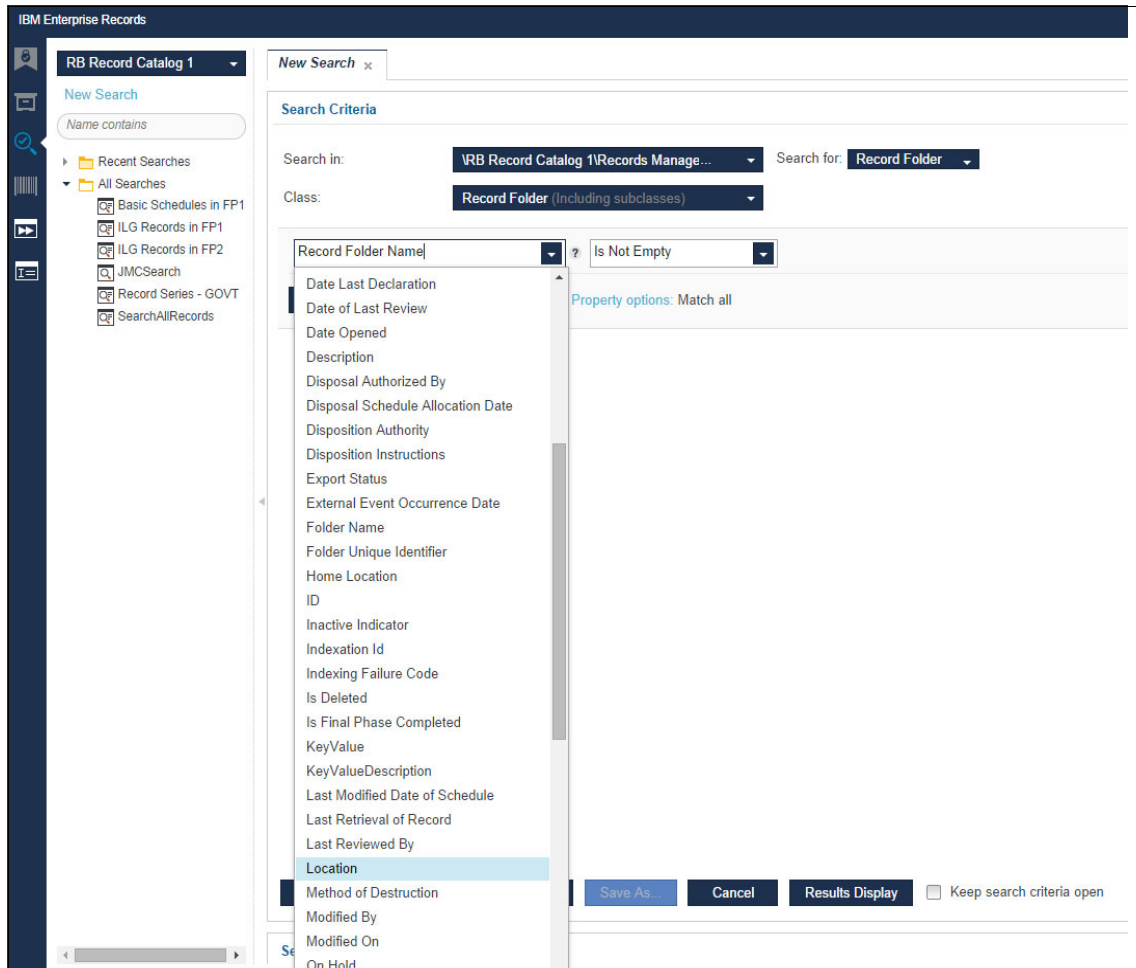


Figure 11-13 Refining the search for boxes, hybrid folders, and physical folders

5. Choose the criteria to display in the search results by selecting **Results Display**. In the Search Results Display view, select the following fields from the Available list:
 - Disposition Authority
 - Disposition Instructions
 - Reviewer
 - Home Location
 - Bar code
6. From the Selected list, use the left arrow to remove the **Modified On** and **Modified By** fields (see Figure 11-14, and then click **OK**.

Search Results Display

Available

- Reason for Outcome of Last Review
- Reason for Relocation
- Record Pattern
- Record Pattern Increment By
- Reopen By
- Reopen Date
- RetentionTriggerDate

Selected

- Record Folder Name
- Disposition Authority
- Disposition Instructions
- Home Location
- Reviewer
- Barcode

Sort by: **Barcode**

Sort order: ☒ Ascending ☐ Descending

☐ Enable content summaries (Text search only)

OK **Cancel**

Figure 11-14 Choosing the results display details

7. When you automatically return to the previous window, shown in Figure 11-12 on page 267, you ready to run the search and view the results, so select **Search**.

Figure 11-15 on page 270 shows the search results.

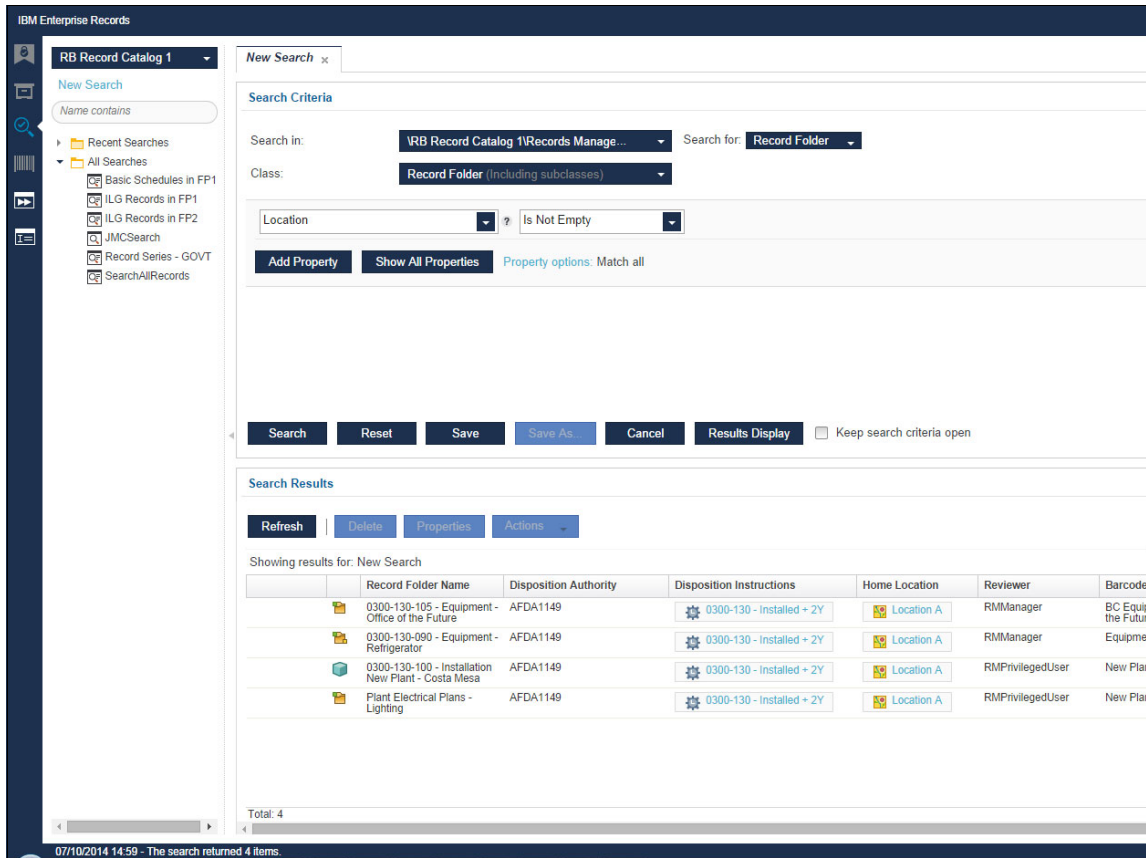


Figure 11-15 Search results looking for boxes, hybrid folders and physical record folders

The search results have identified several physical items, including a box, physical record folders, and a hybrid record folder. By selecting multiple items, the Actions menu becomes active, and you can choose to export the results, which are exported in a comma-delimited format. Figure 11-16 on page 271 shows an export of the search results.

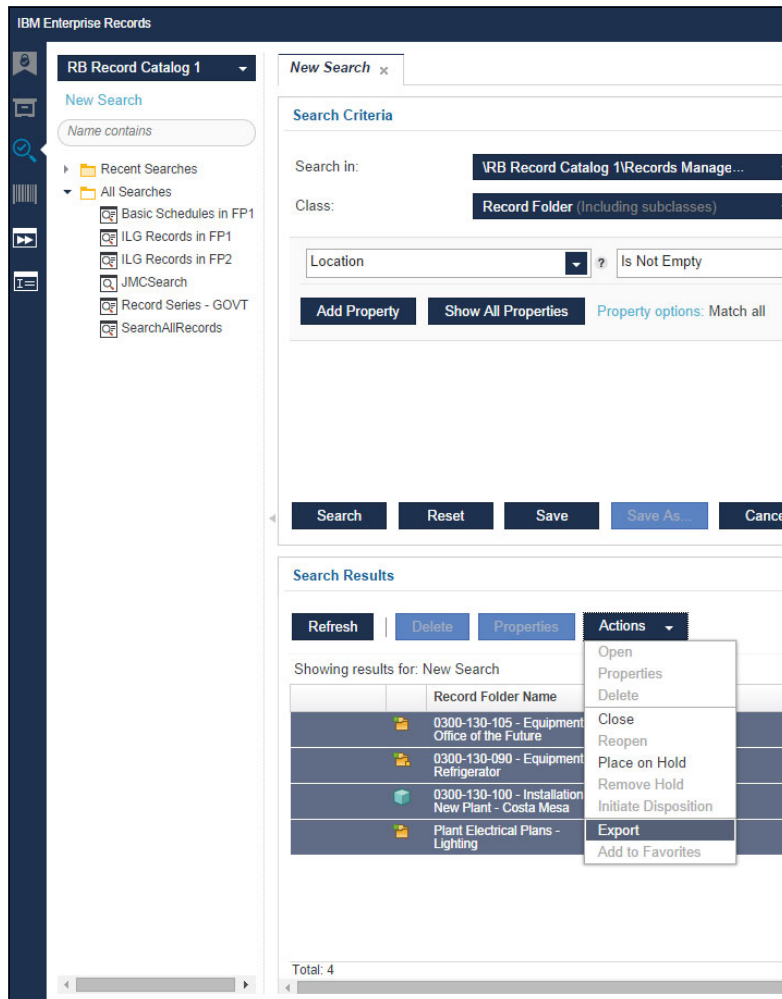


Figure 11-16 Exporting search results

Note: When exporting search results, you can extend or refine the results.

If the search will be used again, you can choose to save the search for reuse. When saving the search for reuse, the subsequent user will not be prompted to re-enter or change any of the search criteria. Selecting the search from the saved searches automatically runs the same search.

In this example, the search results provide a view of physical item information as it was populated in our controlled environment. The values displayed in your environment might differ, largely dependent on how you use these fields.

11.2.5 Reporting

The Search and Export Results feature in Enterprise Records can be used to address rudimentary reporting on physical items.

Although Enterprise Records does not provide preconfigured report templates for physical items, it is possible to create Cognos reports about physical items. For more information, see Chapter 10, “Reporting” on page 243.

11.2.6 Auditing

To enable auditing on physical items in Enterprise Records, it is necessary to configure auditing on the relevant physical items classes for your organization in the file plan object store (FPOS). Enabling auditing is typically done by an administrator of your system, using the administrative console for Content Platform Engine. After configured, changes and updates to physical items appear in the History tab of the item, under the Update Events option in the Enterprise Records desktop.

For more information about auditing, see Chapter 9, “Audit requirements” on page 235.

11.3 Tracking physical records

Enterprise Records uses the workflow features of the Content Platform Engine to handle the processing requirements for managing physical items.

11.3.1 Workflow subscriptions for physical records management

The physical records management workflow is launched through workflow subscriptions. Use the following steps to create a workflow subscription:

1. In the Administration Console for Content Platform Engine (ACCE), select and open your file plan object store from the Object Stores folder.
2. In the tree view, under your Object Store, expand the folders by selecting **Data Design** → **Classes** → **Document** → **Record**.
3. Right-click **Marker Record** and select **New Subscription**.

4. Enter a subscription name, such as Marker Subscription, and click **Next**.
5. Under Subscription Behavior, check **Create a workflow subscription**, and click **Next**.
6. Click **Next** in the Select Triggers window, because no triggers are used.
7. Under Workflow Information, Workflow definition, select the **Physical Record Management** workflow. Check **Allow manual launching**, and click **Next**.
8. Click **Next** in the Build Property Map window.
9. Under Additional Options, check **Include subclasses**, and click **Next**.
10. Click **Finish**.
11. Repeat steps 4 to 10 after right-clicking the following Folder classes, and selecting **New Subscription**:
 - **Data Design → Classes → Folder → RM Folder → Record Folder → Hybrid Record Folder**
 - **Data Design → Classes → Folder → RM Folder → Record Folder → Physical Container → Box**
 - **Data Design → Classes → Folder → RM Folder → Record Folder → Physical Record Folder**
 - **Data Design → Classes → Folder → RM Folder → Volume**

11.3.2 Accessing physical records

Within the Enterprise Records desktop, authorized users may be given access to the work view feature, as illustrated in Figure 11-17 on page 274. Within this interface, a user can see three options, depending upon their access privileges:

My Inbox	This provides you access to any work items that require action by you.
Public Inboxes	This provides you access to public work queues that you have been granted access to and which you may act on.
Tracker	This gives you access to a diagrammatic representation of a workflow, which you have been granted access to view, which includes the workflows that you initiated.

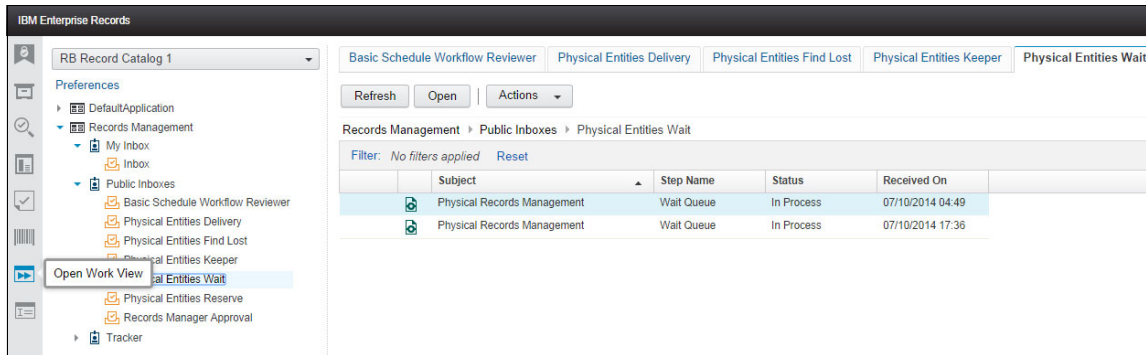


Figure 11-17 Work view interface in an Enterprise Records desktop

To open an item in any of these queues, you have the following options:

- ▶ Select the item and choose the **Open** action from the toolbar.
- ▶ Choosing the **Open** option action toolbar or from the menu on the item.
- ▶ Simply double-click the item.

You can generally find instructions for what you are required to do to act on the item within the Work Item Action interface.

Note: Users must have access to the Open Work View feature in the Enterprise Records desktop to charge out physical items, because those are displayed in their My Inbox views.

11.3.3 Locations, reservations, and charge outs

All physical items in Enterprise Records have two location fields, Home Location and Location, in their metadata. The Home Location field specifies where the item usually is held, and the Location field specifies where the item is held currently. To add or edit locations in Enterprise Records, you need access to the Open Configuration View feature on the desktop, as shown in Figure 11-18 on page 275.

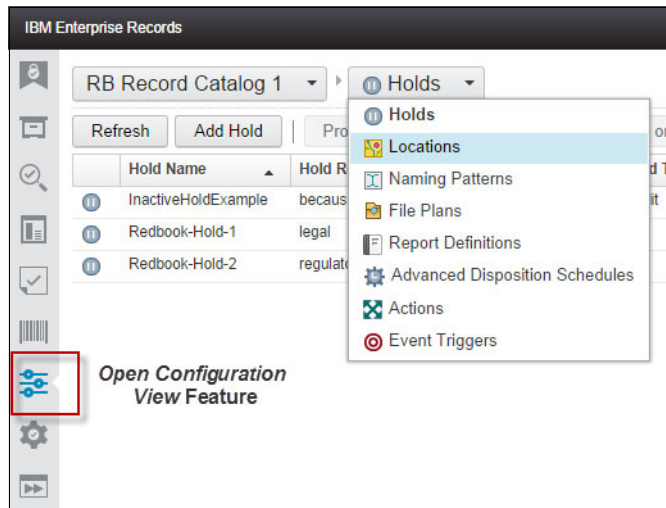


Figure 11-18 Configuration View feature

Access to location maintenance is available from the drop-down menu by selecting the **Locations** menu option. This provides access to the administration feature for Locations, from which you can choose the action to take, such as adding a new location, viewing and editing the properties of a location, or deleting a location. Figure 11-19 shows a snippet of the interface that a user sees when adding a new location

*Location Name: ?	<input type="text"/>
Barcode: ?	<input type="text"/>
Description: ?	<input type="text"/>
Reviewer: ?	RMAdmin

Figure 11-19 Add a new location

The Location profile contains a minimal set of metadata, which might be extended if you have a business requirement for more details to be collected.

Charge outs and reservations

When a user requires access to the item, the user needs to “charge out” the item. This initiates a movement of the item to a new location, which updates the Location value in the physical item’s metadata.

To charge out a physical item from an Enterprise Records desktop, navigate or search for the item in the desktop, select it when you find it, and then click **Charge Out** on the toolbar Actions menu or from the menu for the item. When the interface opens, it shows details about the availability of the item and, if the item is presently charged out, who has the item.

The Charge Out workflow does not involve any overseeing roles. When a request is initiated, a notification is sent to the current holder of the item who can decide whether to release the item to the requester, to hold the item longer (delay), or to report the item as lost. If the user agrees to release the item to the requester, the user must specify how it will be delivered, either manually or through a delivery service.

If it is manually delivered to the receiver, the recipient is notified and must confirm upon receiving the item. After receiving the item, the user who sent the item receives a second notification for release of the item and can nominate how to release the item (manually or through a delivery service). They can also delay release of the item for a period of time, depending on content management rules.

When a delivery service is chosen, the software creates a work item that requires an authorized user to access the Physical Entities Delivery Inbox. The authorized user can then select the **Deliver** option on the interface, and the software sends a notification to a nominated user (recipient) in the new location. The user in the new location can then receive the item.

Another mechanism for processing the notifications for physical items is to use the Open Physical Items View feature, locate the item, and then process the item. This is illustrated in Figure 11-20 on page 277. To act on a task, such as approve item in the Physical Items Workflow pane, select the item, and click **Open** to open the item for processing.

IBM Enterprise Records

Physical Items

RB Record Catalog 1

Entity barcode:

Location barcode: 1

Location description:

Search

Reset

Refresh

Delete

Properties

Actions

Physical Items

	Entity Name	Container ID	Entity Type
	0300-130-090 - Equipment - Refrigerator	0300-130-090	Record Folder
	0300-130-100 - Installation New Plant - Costa Mesa	0300-130-100	Record Folder
	0300-130-105 - Equipment - Office of the Future	0300-130-105	Record Folder
	Plant Electrical Plans - Lighting	0300-130-100 Electrical Plans	Record Folder
	Electrical Plan Lighting Ground floor Costa Mesa Plant a		Record

Physical Item Workflows

Refresh

Open

	Workflow Name	Workflow Step	Requesting User	Request Date
	Physical Records Management	Wait Queue	RMPPrivilegedUser	08/10/2014 07:04
	Physical Records Management	Approve	rmuser	08/10/2014 07:04

Figure 11-20 Working with physical item workflows

Note: Do not modify the Physical Records Management workflow.

If a user requests a physical container item (for example, a box) which contains items that are charged out, the physical container item cannot be charged out.

A *reservation* is a variation of the Charge Out workflow, where you specify a date in the future that you would like to receive the item. When a physical item is reserved, the workflow is launched, and the workflow item is held in a Reservations inbox until the date of the reservation.

If auditing is enabled for the physical item classes in Enterprise Records, updates to the location are recorded in the Update Event entry for the item.

Note: The Physical Records Management Charge Out workflow does not support the selection of multiple items.

Preferable practices typically involve creation of a custom Charge Out workflow that allows users to request multiple items for delivery.



IBM Enterprise Records Java APIs

In this chapter, we describe the IBM Enterprise Records application programming interfaces (APIs), with examples.

We cover the following topics:

- ▶ Introduction to IBM Enterprise Records APIs
- ▶ Java API for Records Manager
- ▶ Records Manager API
- ▶ Bulk Declaration Service
- ▶ Performance considerations

12.1 Introduction to IBM Enterprise Records APIs

IBM Enterprise Records software provides multiple APIs, which are a set of platform-independent application programming interfaces written in Java. They are used to interact and perform records management operations in the Content Platform Engine (CPE), which is the content and workflow-management component of the IBM FileNet P8 platform that serves as the foundation of the Enterprise Records application.

You use the Enterprise Records Java API for these tasks:

- ▶ To customize an application that uses record management functions
- ▶ To develop new functions related to records management

There are three sets of APIs in Enterprise Records:

- ▶ The Java API for Record Manager (JARM)

This API set is the newest one and replaces the RM API. It is based on the Java API for Content Engine (JACE), which is the current Content Platform Engine API.

JARM 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. It provides extensive validation to ensure that Records Manager-related logic is followed for record, file plan, and other Records Manager-related actions.

This is the API to use for all records management actions.

- ▶ The Records Manager API (RM API) (deprecated)

This is the previous Record Manager API. It uses version 3.5 of the Content Java API (for Content Engine operations). Because this Content Engine API is progressively phased out, the RM API should not be used anymore. It is provided only for compatibility with an earlier version, in case you have existing custom applications based on this API. Any new development should use JARM.

- ▶ The Bulk Declaration Service (BDS)

The BDS API is limited in scope to performing record declaration in bulk. JARM and RM API declare documents as records one at a time. That presents performance-related limitations when declaring thousands of records. The BDS API introduces bulk declaration.

Rather than passing one record definition to the API, the client application can pass an area of records. The BDS uses the Content Platform Engine batch process to process all of those records in a single transaction. This is the preferred API when declaring a large number of records.

12.2 Java API for Records Manager

The Java API for Records Manager (JARM) requires the Java 2 Standard Edition (J2SE) 1.6 or later development environment. This is the newest API set, and it should be used for any new development.

For more information about JARM, see the following documentation:

- ▶ “Developing with New IBM Enterprise Records Java API,” a PDF file on the server where Enterprise Records is installed, either [/Program Files] or [/opt]\EnterpriseRecords\API\JARM
- ▶ For sample code that uses JARM to declare documents as record or to use it in the context of a Content Engine event, see Chapter 18, “IBM Java API for Records Manager case study” on page 385.

12.3 Records Manager API

The Enterprise Records Java API requires the Java 2 Standard Edition (J2SE) 1.4.2 or later development environment. When using the Records Manager API (RM API), the following .jar files must be included in your class path:

- ▶ javaapi.jar
- ▶ pe.jar
- ▶ rmap.jar
- ▶ rmapresources.jar
- ▶ rm-bds.jar
- ▶ accessrole.jar
- ▶ Jace.jar

Note: This API has been replaced by the newer JARM API. It is still delivered for compatibility with an earlier version, but use JARM instead for new development.

12.4 Bulk Declaration Service

Bulk Declaration Service (BDS) is a mechanism to perform multiple record operations in batches. BDS is implemented by 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 FileNet Content Engine
- ▶ Bulk declaration of physical records
- ▶ Bulk creation of new FileNet Content Engine documents and declaring them as records
- ▶ Bulk creation of new Content Engine documents

Most BDS operations can be performed by using Enterprise Records APIs also, but BDS is much faster because of the batch operation.

Using BDS adds an extra load on the system, of course. If you have only a few records to declare at a time, we suggest using Enterprise Records APIs. If you must declare a high volume of records, we suggest 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.
4. Check batch results.

The power of BDS lies in batching, which results in saving 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 by using the BDS API to declare records quickly.

For more information about BDS interfaces and classes, see the online `ecm_help` documentation.

12.5 Performance considerations

Following these suggestions will improve performance:

- ▶ While querying Content Platform Engine for data, retrieve only the minimum data required to complete the task. Minimize data retrieval in the following ways:
 - If executing a Content Engine CE SQL statement, select only the necessary columns.
 - If executing a `getObject()` request, use a property filter and request only the necessary properties.
 - Use paging when the result set can be large. Returning a large result set can use too much memory.
 - If possible, process the result set immediately and do not save the data for later use. This ensures that memory use 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 used 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, limit the batching according to the resources available. Balanced batching results in high performance.



IBM Enterprise Records for IBM Content Manager

This chapter presents the redesigned IBM Enterprise Records for IBM Content Manager that provides direct access to external repositories, such as Content Manager V8, without the need to use federation.

This chapter covers the following topics:

- ▶ Presentation
- ▶ Architecture
- ▶ Difference with Content Federation Services
- ▶ Java for Records Manager

13.1 Presentation

Enterprise Records has been redesigned to be more efficient to more tightly integrate with IBM Content Manager and use the native Content Manager capabilities to manage the records in place. This enables you to add compliant records management functions without affecting your existing applications and users. Existing Content Manager customers can now apply their investment in this scalable enterprise content management repository to manage records also.

Enterprise Records enables end-to-end lifecycle management of Content Manager content records, all the way from declaration, to approval workflows, to disposition, with compliant audit logging and reporting. Organizations that are able to execute records management policies on their Content Manager repositories will benefit by reducing noncompliance risk and reducing costs associated with over-retention of information in the Content Manager repository.

Routinely disposing of unnecessary data in the Content Manager repository also improves the performance of the Content Manager system and the relevance of search results and analytics.

Note: IBM Enterprise Records for Content Manager V8 replaces Content Federation Services (CFS) and IBM Records Manager.

13.2 Architecture

To improve performance, Enterprise Records now adds direct access to external repositories without the need for federation. This reduces the P8 footprint, as shown in Figure 13-1 on page 287. This redesign incorporated these key elements:

- ▶ The record's metadata is no longer stored in the Content Process Engine. It is now in an external database and optimized for high performance.
- ▶ Content Manager is improved with a bulk API to improve record declaration and destruction performance.
- ▶ As of IBM Enterprise Records 5.2.0 Fix Pack 1 (FP1), the file plan is still stored and managed in a file plan object store repository in Content Process Engine.
- ▶ There is no need to use federation to declare externally managed documents as records. Enterprise Records connects natively to the external documents.

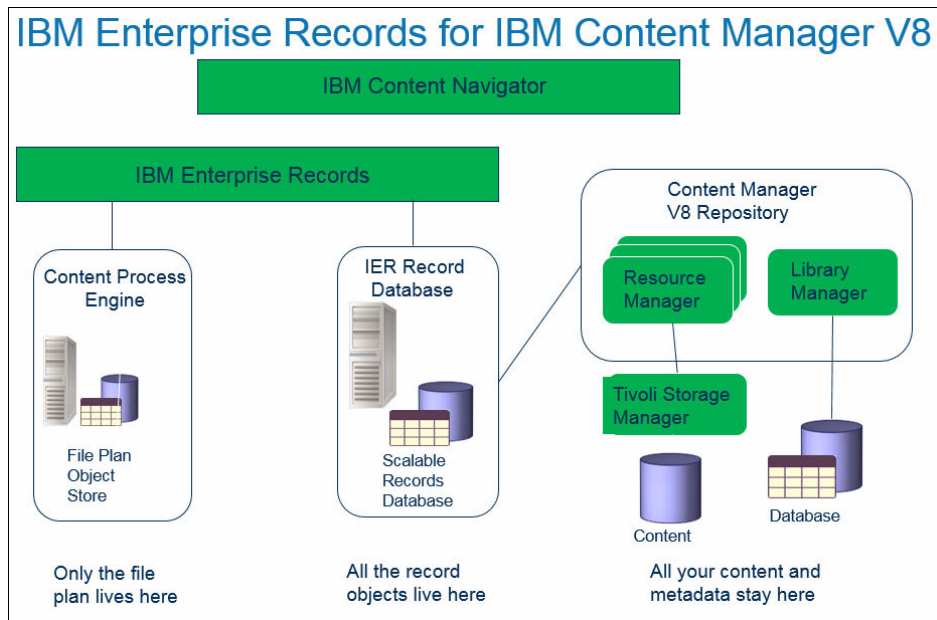


Figure 13-1 IBM Enterprise Records for IBM Content Manager V8

13.3 Difference with Content Federation Services

The current solution to support a non-P8 content is Content Federation Services (CFS). This requires a separate federation process that duplicates the documents' metadata from the external repository to a P8 repository and then declares the duplicated P8 document as record. Tracking these documents causes a high level of indirection and overhead that leads to lower performance. Moreover, CFS relies heavily on P8.

Unlike CFS, Enterprise Records for Content Manager provides direct support to a Content Manager repository, without the need for federation. This also minimizes the P8 footprint.

The record declaration and disposition processing performance is strongly improved over the existing P8 Content Engine and CFS-based solution

Some of the main differences are highlighted in the Table 13-1.

Table 13-1 Differences between Enterprise Records for Content Manager and CFS

IBM Enterprise Records for Content Manager	Content Federation Services
Direct connection to Content Manager.	Federation enables you to connect to many source repositories, including Content Manager.
Record lockdown is performed by using a user-defined super user account.	A special user is used to lock down content.
Record security is controlled from the record repository. Native Content Manager security on a document is retained, except deletion.	Same security.
Typical Content Manager and Enterprise Records installation.	This is difficult to set up and tightly coupled with IBM Content Integrator.
Administration is integrated with Enterprise Records plug-in on top of IBM Content Navigator.	Has its own administrative web application.
Database is easy to configure and tune.	Database configuration is automatic when creating a CFS Fixed Content Device (FCD).
Content is stored natively in Content Manager.	Content is normally stored natively in Content Manager but can be moved to P8.
No duplication of the metadata.	Metadata is duplicated in a P8 object store and requires a synchronization process if it changes in the Content Manager repository.
Higher performance.	Lower performance.

13.4 Java for Records Manager

Java for Records Manager (JARM) is updated to support this new feature. Except for the initial connection to repositories, all API calls are the same, regardless of whether you connect to a regular P8 record object store or to a Content Manager repository.

Example 13-1 is JARM sample code that connects to an Enterprise Records for Content Manager V8 repository.

Example 13-1 JARM sample code

```
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import javax.security.auth.Subject;
import com.ibm.jarm.api.constants.DomainType;
import com.ibm.jarm.api.constants.EntityType;
import com.ibm.jarm.api.constants.RMPPropertyName;
import com.ibm.jarm.api.core.DomainConnection;
import com.ibm.jarm.api.core.RMDomain;
import com.ibm.jarm.api.core.RMFactory;
import com.ibm.jarm.api.core.SRM_RMR_ConnectionInfoEntry;
import com.ibm.jarm.api.core.SRM_R_ConnectionInfoEntry;
import com.ibm.jarm.api.util.RMUserContext;
import com.ibm.jarm.api.core.RMFactory.FilePlanRepository;
import com.ibm.jarm.api.core.RMFactory.ContentRepository;
import com.ibm.mm.sdk.common.DKDDO;
import com.ibm.mm.sdk.server.DKDatastoreICM;
import com.ibm.jarm.api.core.ContentItem;
import com.ibm.jarm.api.core.RecordContainer;
import com.ibm.jarm.api.property.RMPProperties;

...

// P8 Content Engine connection information
String protocol      = "http";
String ceServerName  = "<ceServerName>";
String cePort        = "<cePort>";
String ierUserName   = "<ierUserName>";
String ierPassword   = "<ierPassword>";
String JAAS_STANZA   = RMUserContext.P8_STANZA_WSI;

// Record database connection information
String r_DBName      = "<r_DBName>";
String r_DBHost      = "<r_DBHost>";
String r_DBPort      = "<r_DBPort>";
```

```

String r_DBUserName = "<r_DBUserName>";
String r_DBPassword = "<r_DBPassword>";

// P8 File Plan Object Store connection information
String      fposName = "<FPOSName>";

// Content Manager V8 datastore connection information
String cm8ServerName = "<cm8ServerName>";
String cm8UserName   = "<cm8UserName>";
String cm8Password   = "<cm8Password>";

// Get a CM8 datastore instance
DKDatastoreICM cm8DataStore = new DKDatastoreICM();
cm8DataStore.connect(cm8ServerName, cm8UserName, cm8Password, "");

// Instantiate the connectionInfo hash map
Map<String, Object> connectionInfo = new HashMap<String, Object>();

// Define the R (new Record manager application) connection information to the
RMR database
// This is the database where the records are stored
Map<String, SRM_R_ConnectionInfoEntry> rParameters = new HashMap<String,
SRM_R_ConnectionInfoEntry>();
rParameters.put(fposName, new SRM_R_ConnectionInfoEntry(r_DBName, r_DBHost,
r_DBPort, r_DBUserName, r_DBPassword));
connectionInfo.put(DomainConnection.KEY_SRM_R_MAP, rParameters);

// Define the RMR connection information to CM8 datastore
// This is where the CM8 repository
Map<String, SRM_RMR_ConnectionInfoEntry> rmrParameters = new HashMap<String,
SRM_RMR_ConnectionInfoEntry>();
rmrParameters.put(cm8DataStore.datastoreName(), new
SRM_RMR_ConnectionInfoEntry(cm8DataStore));
connectionInfo.put(DomainConnection.KEY_SRM_RMR_MAP, rmrParameters);

// Create the CE domain connection and set the subject
String url = protocol + "://" + ceServerName + ":" + cePort +
"/wsi/FNCEWS40MTOM";
DomainConnection jarmDomainConnection =
RMFactory.DomainConnection.createInstance(DomainType.P8_SRM, url,
connectionInfo);
RMUserContext uc = RMUserContext.get();
Subject subject = RMUserContext.createSubject(jarmDomainConnection,
ierUserName, ierPassword, JAAS_STANZA);
uc.setSubject(subject);

// Instantiate a jarmDomain

```

```

RMDomain jarmDomain =
com.ibm.jarm.api.core.RMFactory.RMDomain.fetchInstance(jarmDomainConnection,
null, null);

// Instantiate a JARM FPOS repository and a JARM Content Repository
jarmFpos = FilePlanRepository.fetchInstance(jarmDomain, fposName, null);
jarmCM8Repository = ContentRepository.fetchInstance(jarmDomain,
cm8DataStore.datastoreName(), null);

```

After the connection to the file plan object store and to the ContentRepository is established, JARM can declare a Content Manager document as a record, as shown in Example 13-2

Example 13-2 Declaring a Content Manager document as a record

```

public void declareCM8DocumentAsRecord(DKDDO cm8Document, String cm8RecType,
RecordContainer recordCategory)
{
    // Instantiate the list of CM8 documents
    List<ContentItem> contentItemsToDeclare = new ArrayList<ContentItem>();
    ContentItem ci =
RMFactory.ContentItem.getInstanceFromObject(jarmCM8Repository, cm8Document);
contentItemsToDeclare.add(ci);

    // Prepare the record metadata properties
    RMProperties jarmProps =
com.ibm.jarm.api.core.RMFactory.RMProperties.createInstance(com.ibm.jarm.api.co
nstants.DomainType.P8_SRM, EntityType.Record);
    jarmProps.putStringValue(RMPropertyName.DocumentTitle, "Record Title");

    // Declare the document as record into the IER category "recordCategory"
    recordCategory.declare(cm8RecType, jarmProps, null, null,
contentItemsToDeclare);
}

```



Part 2

Implementation case studies

In this part, we use a case study to provide step-by-step instructions for implementing a sample records management solution. We give concrete examples of how to perform the tasks, including file plan creation, records ingestion and declaration, record disposal, record hold, and sample programs that use IBM Enterprise Records application programming interfaces (APIs).



File plan case study

In this chapter, we describe the steps required to create a file plan in IBM Enterprise Records.

We cover the following topics:

- ▶ Types of object stores
- ▶ File plan case study introduction
- ▶ Creating a file plan in IBM Enterprise Records

Note: This chapter does not provide the details of installing Enterprise Records or the details of performing the standard Content Platform Engine functions, such as creating object stores and defining document classes. For these instructions, see the software's online help, `ecm_help`.

14.1 Types of object stores

For Enterprise Records, a record is stored as metadata. *Metadata* is data about data. It's a file that references and contains information about another electronic file (document) or physical object.

- ▶ Electronic files

An electronic file or document might be a single file, a digital photo, or a set of related files that can be treated as one object. For example, a set of files is an email message and its attachments. Electronic files exist in object stores and other repositories. When you declare the document as a record, Enterprise Records manages the document, including its security and possible classification. This security setting can change the access to the document. Therefore, the author of the document can be prevented from changing the document.

- ▶ Physical objects

Physical objects might be audio tapes, video tapes, microfilm, hard disks, DNA samples, printed paper documents, and photos. They might be stored in boxes or file cabinets inside warehouses and other secure archival facilities. The record tracks the location of the physical object.

For Enterprise Records configuration, there are two types of object stores, which we describe in the subtopics that follow:

- ▶ File plan object store (FPOS)
- ▶ Record-enabled object store (ROS)

14.1.1 File plan object store

Objects that are declared as records are stored in a repository that is called a *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 that are required to manage records. When documents in the ROS (see next type) are declared as records, the record-related information (metadata) is stored as a separate record object in the FPOS.

Terminology: A *file plan object store* contains a file plan, which is a hierarchy of record management objects that are needed to classify records.

14.1.2 Record-enabled object store

A repository that contains documents that can be declared as records is called a records repository or *record-enabled object store*.

The record-enabled object store (ROS) serves as the content repository for electronic documents. Documents stored in an ROS can be declared as records.

Terminology: A *record-enabled object store* contains documents that you can declare as records.

14.2 File plan case study introduction

Our case study is based on the fictitious global financial institution XYZ that we introduced in 3.6, “Case study: File plans in IBM Enterprise Records” on page 90.

The part of the company’s file plan that includes categories that are related to the case study for this book is illustrated in Figure 14-1. This file plan is designed to showcase the following:

- ▶ A variety of ingestion and record declaration options that Enterprise Records offers
- ▶ The impact of various disposition aggregation levels on disposition and hold processing

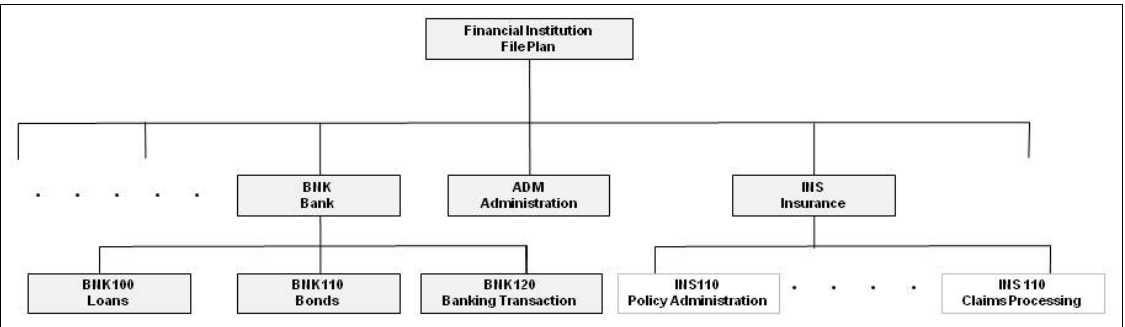


Figure 14-1 Partial file plan showing categories related to the case study for this book

The case study focuses on three areas of the file plan:

- ▶ Employee data:
 - Record level aggregation
 - Event-based retention rule applies
- ▶ Broker-dealer transaction:
 - Record level aggregation
 - Fixed retention rule applies
- ▶ Corporate governance:
 - Permanent retention rule applies

For performance, backup, and recovery reasons, the preferred practice is to have the FPOS and ROS in separate object stores. For this case study, we create two object stores:

- ▶ RB Record Catalog 1 (FPOS)
- ▶ RB Content Repository 1 (ROS)

Records enabling in a IBM FileNet P8 system requires a thorough understanding of IBM FileNet Enterprise Manager and the IBM Enterprise Records application. This chapter does not provide installation instructions for the IBM FileNet P8 products. We assume that you have a working version of Enterprise Records. The steps in this chapter are required and must be performed each time that you need to create a new set of object stores for use with Enterprise Records.

14.3 Creating a file plan in IBM Enterprise Records

There are different ways of setting up file plans in Enterprise Records.

A Records Retention Policy can be implemented in the Policy Management module (Global Retention Policy and Schedule Management). The master schedule and local schedules are based on certain criteria, for example, geographical locations can be syndicated to Enterprise Records.

Another way of implementing a file plan is to enter it directly into Enterprise Records. Our case study shows how to create a file plan from Enterprise Records.

In our case study of a global financial institution, the retention rules for a multinational financial institution take varying retention requirements for records into consideration.

In Enterprise Records, a basic disposition schedule is used to illustrate the record-level aggregation. *Record Category* is used to represent the implementation of a functional level, such as accounting, broker-dealer transactions, corporate governance, insurance, and annuity in our example.

Note: A *basic disposition schedule* is a high-performance schedule that is intuitive to use.

Within each category, records can be broken down further into smaller groupings. A basic schedule with immediate Destroy without administrator review is shown in the case study. Records are destroyed after the review period expires for an event-based client.

This next section shows the steps for creating and populating a file plan.

14.3.1 Create a new file plan

Complete the following steps to create a new file plan:

1. Log in to Enterprise Records as an authorized user.
2. Click **Open Configuration View** from the shortcut menu on the left, as shown in Figure 14-2.

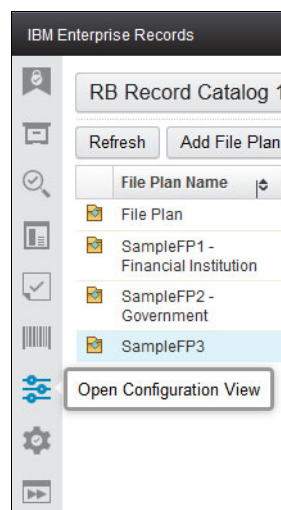


Figure 14-2 Select Open Configuration View

3. Select the repository where the object store resides.
4. Select the **File Plans** view, as shown in Figure 14-3 on page 300.

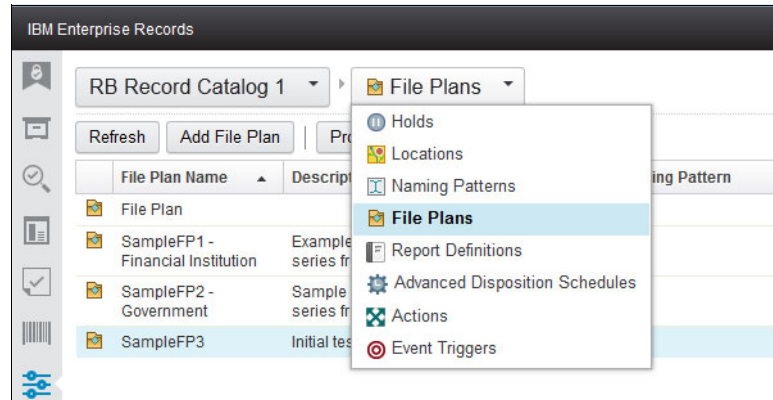


Figure 14-3 Select File Plans view

5. Click **Add File Plan**, as shown in Figure 14-4.

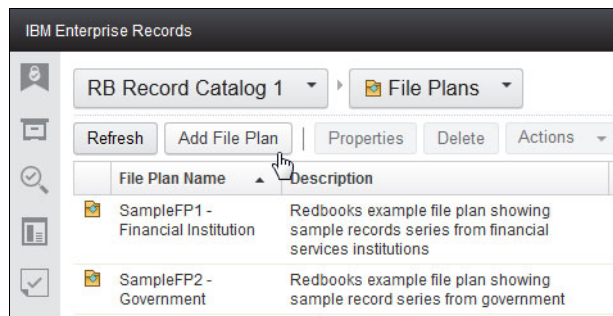


Figure 14-4 Add File Plan tab

6. Complete the details as shown in Figure 14-5 on page 301. Enter the file plan name, a description of the file plan, security, and (optionally) a naming pattern.

Note: *Naming patterns* are a way to ensure consistency in naming the record category names and IDs, but we do not use them in this book.

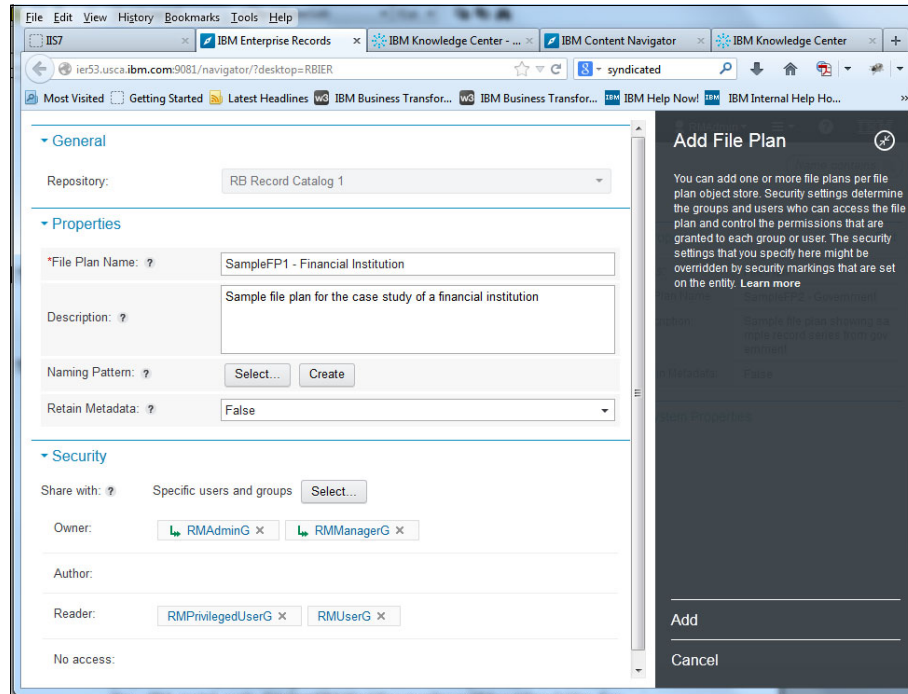


Figure 14-5 Adding a new file plan

7. In the right panel, click **Add** to create the new file plan.

14.3.2 Browse the file plan

To browse a file plan, follow these steps:

1. Continue from the previous step.
2. Click **Browse File Plan** from the shortcut menu at the left, as shown in Figure 14-6 on page 302.

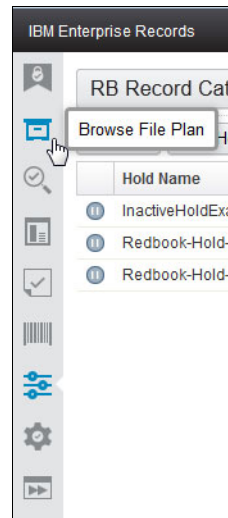


Figure 14-6 Browse File Plan

3. Click the **Select File Plan** icon, and pick the file plan that was created, as shown in Figure 14-7 on page 303.

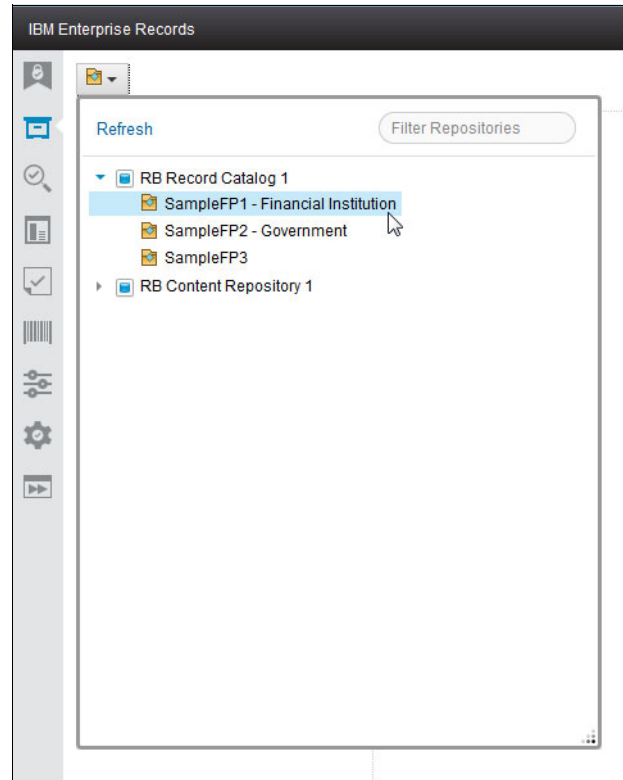


Figure 14-7 Select File Plan

4. You can now start populating the file plan.

14.3.3 Populate the file plan

Having created the file plan, you now need to build the file plan categories. We create a hierarchy of record categories that maps to Figure 14-8 on page 304.

Terminology: A *record category* is a container that categorizes a set of related records within a file plan. You typically use record categories to classify records based on functional categories.

A record category can contain subcategories. The category has a name and an ID. Both display in the path, separated by a hyphen. Both the name and the ID must be unique within the parent container.

In our example, we create a record category called “EMP - Employee Data” and two subcategories:

- ▶ EMP100 - Compensation
- ▶ EMP110 - Employee identity under the category.

Record Series (Primary)	Record Series Description	Citation and time	Retention Period	Record Sub-Series/ Secondary Classification	Record Sub-Series Description
Employee Data	Monitoring and reporting on all employee related activity and activity of associated persons such as: Employee		Employment + 3	Employee Files	Documents containing core employee information, performance reviews,
		17cfr240.17a-3(a)(12)(i), employment +3yrs		Employee Identity and Security	Documents pertaining to fingerprinting, criminal checks, identify verifications
		17cfr240.17a-3(a)(12)(i)(A), employment +3yrs			
		17cfr240.17a-3(a)(19)(i), 3yrs		Compensation/Salary	Documents pertaining to employees salaries, hours worked, gross wage,
		17cfr240.17a-3(a)(19)(ii), 3yrs		Commissions/Bonuses	Documents pertaining to commission or bonus information including allocations
		17cfr240.17f-2(d) employment +3yrs			
		29CFR516.2		Insurance Certification	Licensing and certification from State Insurance Bodies and other related
		29CFR516.5		Registration	Licensing applications and renewals documentation, terminations,
				Affirmative Action	Documents containing information about employee diversity, EEO
				Personal Securities Transactions	Documents pertaining to employee sale/transfer/disposition of owned

Figure 14-8 Employee data

To populate a file plan, follow these steps:

1. Log in to Enterprise Records as an authorized user.
2. From the shortcut icons at the left, select **Browse File Plan**.
3. Click **Select File Plan** and pick the file plan that you created. In this example, that is: SampleFP1 - Financial Institution.
4. Click **Add Record Category**, as shown in Figure 14-9.

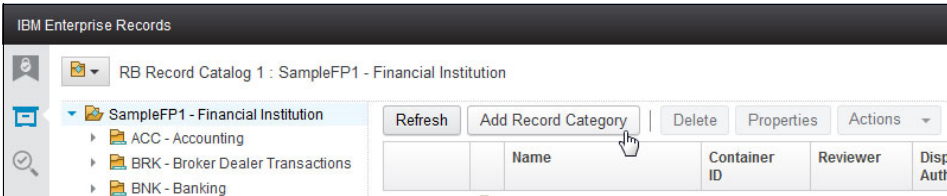


Figure 14-9 Add Record Category

5. Complete the fields shown in Figure 14-10 on page 305, including Record Category Name, Record Category Identifier, Description, and Disposition. In this case, because there is no retention rule applies, *no schedule* is selected.
6. Click **Add**.

Figure 14-10 Setting up first level

7. Repeat the previous step to set up the rest of the first-level record categories.
8. Now you can create the record categories on the second level. This is the level which retention rules are applied.
9. From the File Plan view, select the first-level record category and click **Add Record Category** (Figure 14-11).

Figure 14-11 Setting up second-level record categories

10. Complete the details, including Record Category Name, Record category Identifier, Description, and Disposition.

In this example, we demonstrate the use of a basic disposition schedule. The retention rule for Employee Data - Compensation is three years after the termination of employment. To set up an event-based disposition schedule, under Disposition, select **RetentionTriggerDate** as the retention trigger property name and **3** years as the retention period. See Figure 14-12.

The screenshot shows the 'Add Record Category' dialog box in the IBM Enterprise Records application. The main form has the following fields:

- Record Category Name:** EMP100 - Compensation
- Record Category Identifier:** EMP100
- Description:** Documents pertaining to employees salaries, hours worked, gross wage, deductions
- Date Opened:** 9/23/2014, 4:04 PM
- Reviewer:** RMAAdmin

Under the **Disposition** section, the **Basic disposition schedule** radio button is selected. The **Retention trigger property name** is set to **RetentionTriggerDate**, and the **Retention period** is set to 3 Years, 0 Months, and 0 Days.

The right-hand sidebar shows a list of record categories, including 'EMP - Employee Data', 'EMP', and 'EMP - Employee Data'. At the bottom of the sidebar, there are 'Add' and 'Cancel' buttons.

Figure 14-12 Second-level record category

11. Click **Add**.

Figure 14-13 shows how the file plan looks like for the Employee Data record category.

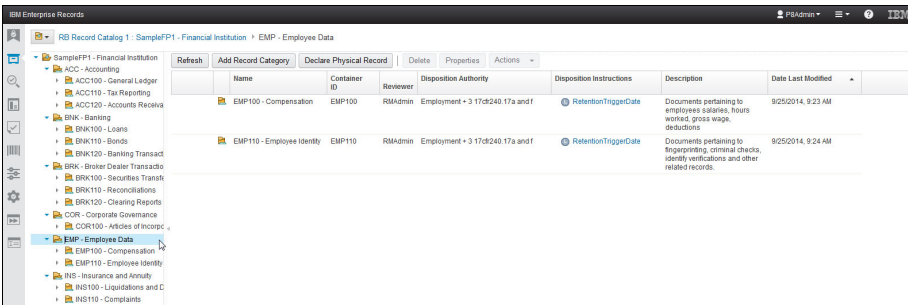


Figure 14-13 Employee data

12.Repeat steps 7 to 9 to set up the rest of the second-level record category. For a fixed disposition schedule, under Disposition, select **Documented** as the retention trigger property name and **6** years as the retention period. See Figure 14-14.

Basic disposition schedule

* Retention trigger property name ? DocumentDate

* Retention period ? 6 Years 0 Months 0 Days

Figure 14-14 Fixed retention period

13.For records with a permanent retention period, under Disposition, select **No schedule** (see Figure 14-15).

☒ No schedule

Figure 14-15 No Schedule

14.Click **Add**.

Figure 14-16 shows what the partial file plan looks like.

The screenshot displays the IBM Enterprise Records interface. On the left, a tree view shows the hierarchy of record categories under 'SampleFP1 - Financial Institution'. The main pane shows a table of records with columns: Name, Container ID, Reviewer, Disposition Authority, Disposition Instructions, Description, and Date Last Modified. The table lists several record categories including Accounting, Banking, Corporate Governance, Employee Data, Insurance and Annuity, and Broker Dealer Transactions.

Name	Container ID	Reviewer	Disposition Authority	Disposition Instructions	Description	Date Last Modified
ACC - Accounting	ACC	RAdmin			Financial Statements, Journal Entries, General Ledger, P&L, Registers, Fixed Assets, Employee Payroll Accounting, Payroll Registers, Tax Returns and Working Papers, Checks, Bank Deposits, Cancelled Checks, Bills	9/24/2014, 3:07 PM
BNK - Banking	BNK	RAdmin			Banking records	9/24/2014, 3:07 PM
COR - Corporate Governance	COR	RAdmin			Books and records that substantiate the existence, operation and obligations of each legal entity, such as: Articles of Incorporation, Board of Directors, Shareholders, Committees, Directors Compensation	9/24/2014, 3:08 PM
EMP - Employee Data	EMP	RAdmin			Monitoring and reporting on all employee related activity and activity of associated persons such as: Employee Files, Fingerprinting, Compensation, Benefits, Registration & Licensing	9/24/2014, 3:08 PM
INS - Insurance and Annuity	INS	RAdmin			Insurance and annuity applications and accompanying records, Product administration records	9/24/2014, 3:09 PM
BRK - Broker Dealer Transactions	BRK	RAdmin			Securities Records; Blotters; Ledgers; Trade Confirmations; Instructions; Client Statements; Reconciliations; Trade Sheets; Stock Bond Records; Money Balance Positions; Asset and Funds Transfer; Receipts	9/24/2014, 3:29 PM

Figure 14-16 Partial file plan of a sample financial institution

The file plan is now populated.



Basic disposition case study

This chapter provides a case study for using basic disposition schedules, scheduling and running the basic disposition sweep, and processing the basic disposition workflow.

This chapter describes the following tasks:

- ▶ Create a new record category with a basic disposition schedule
- ▶ Schedule a basic disposition sweep for report only
- ▶ Schedule a basic disposition sweep for immediate destruction
- ▶ Schedule a basic disposition sweep for approval before destruction

15.1 Create a new record category with a basic disposition schedule

In this section, we demonstrate how to add a new record category with a basic disposition schedule.

In this scenario, ADM120 is a record category in our sample file plan that contains the subcategories for each organization unit using this record series, as shown in Figure 6-14 on page 173 where the parent record category represents the record series.

Figure 15-1 shows this file plan structure with three basic schedules that already exist for three of the business units using ADM120. We will add a fourth category for a new business unit.

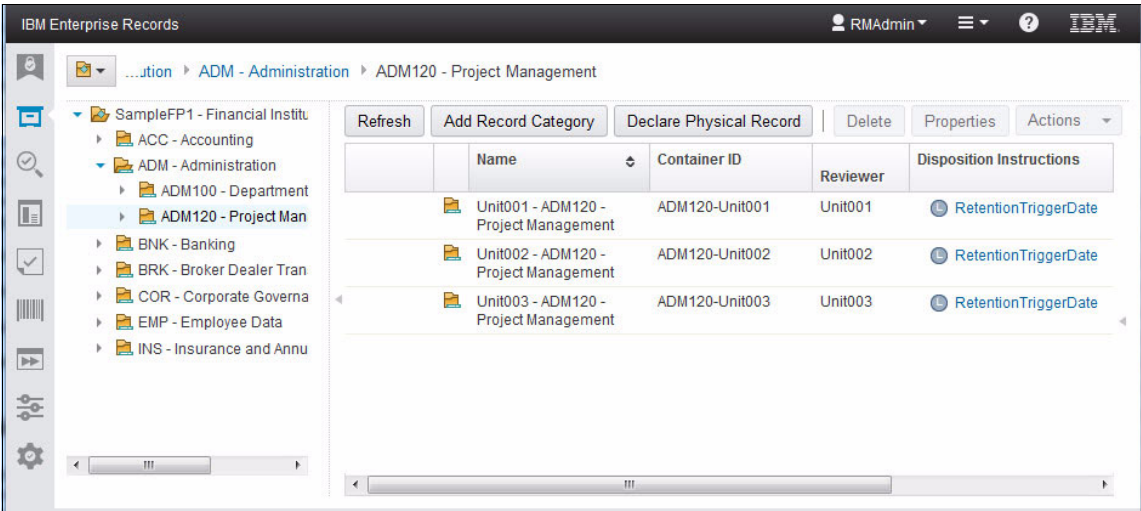


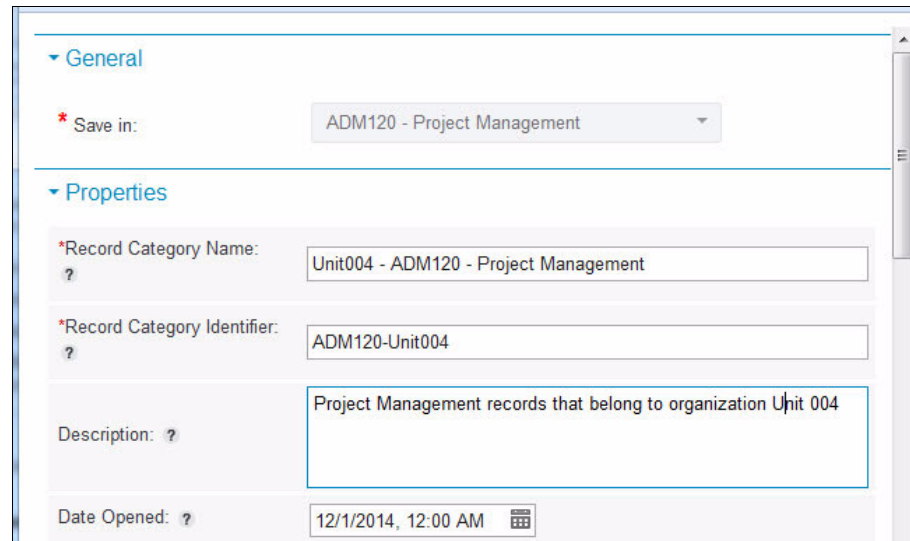
Figure 15-1 ADM120 record category has one basic schedule category for each business unit

In this case study example, we add a record category with a basic schedule to the parent record category ADM120.

To add the record category, follow these steps.

1. Browse to the **ADM120 - Project Management** record category.
2. Click **Add Record Category**.

3. Complete the properties pertaining to this record category, such as Record Category Name, Record Category Identifier, and Description as shown in Figure 15-2. In this example, both the Record Category Name and Record Category Identifier use a standardized naming convention that indicates the record series and which business unit is responsible for the records in that category.



The screenshot shows a web-based form for configuring a record category. It is divided into two main sections: 'General' and 'Properties'. In the 'General' section, there is a 'Save in:' dropdown menu currently set to 'ADM120 - Project Management'. The 'Properties' section contains four fields: 'Record Category Name' with the value 'Unit004 - ADM120 - Project Management', 'Record Category Identifier' with the value 'ADM120-Unit004', 'Description' with the value 'Project Management records that belong to organization Unit 004', and 'Date Opened' with the value '12/1/2014, 12:00 AM'. Each field in the 'Properties' section has a small question mark icon to its left, indicating it is a required field. The 'Date Opened' field includes a calendar icon for date selection.

Figure 15-2 Record category name and record category identifier are required properties for a new record category

4. Under Disposition (Figure 15-3), select **Basic disposition schedule** to fill in the required properties.

▼ Disposition

Basic disposition schedules are high-performance schedules that are easy to use. Advanced disposition schedules offer more capabilities and flexibility, but they are more complex and negatively impact performance.

☐ No schedule

☒ Basic disposition schedule

* Retention trigger property name ?

* Retention period ? Years Months Days

☐ Advanced disposition schedule

* Disposition instructions:

Disposition authority: ?

Figure 15-3 Configuring the required basic disposition schedule properties

Select the appropriate retention trigger property name from the drop-down menu and fill in the appropriate retention period values for Years, Months, and Days. as shown in Figure 15-3. In this example, we select **RetentionTriggerDate** and configure a retention period of 3 years.

Note: The `RetentionTriggerDate` property is a custom date property that was added to the default data model as part of the case study object store configuration. This property was added to a valid record class definition, making it available for use with Enterprise Records.

5. Click **Add** to create the new record category.

The new record category appears in the file plan with the basic schedule configured, as shown in Figure 15-4 on page 313.

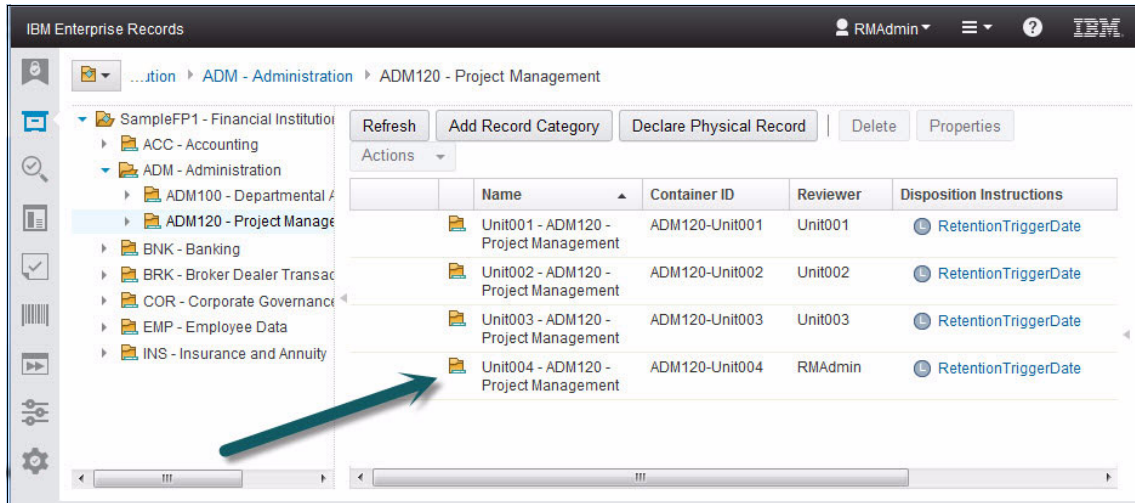


Figure 15-4 A new record category is added to the file plan with a basic disposition schedule

15.2 Schedule a basic disposition sweep for report only

In this section, we demonstrate how to schedule a basic disposition sweep to generate retention due reports for the record series ADM120 without running any disposition process.

In this scenario, we want to generate the retention due reports for each of the organization units that have records in the ADM120 record series. We want to see which records will be due for disposition 30 days from today (the day we run the sweep).

15.2.1 Schedule the sweep

To schedule a basic disposition sweep for a report only, follow these steps:

1. Open **Tasks View** from the Enterprise Records desktop.

2. From the Schedule menu, select **Schedule Basic Disposition Sweep**, as shown in Figure 15-5.

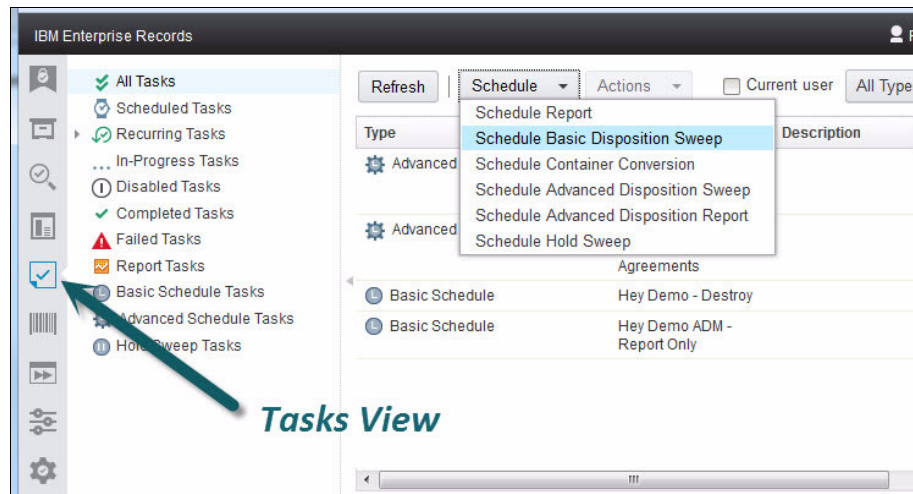


Figure 15-5 Schedule a basic disposition sweep from the Tasks view

3. Select the parameters for the basic disposition sweep, as shown in Figure 15-6. In this case, we select the file plan repository, the ADM120 records category for containers for sweep, a report review period of **30** days, and set Report Only to **Yes**.

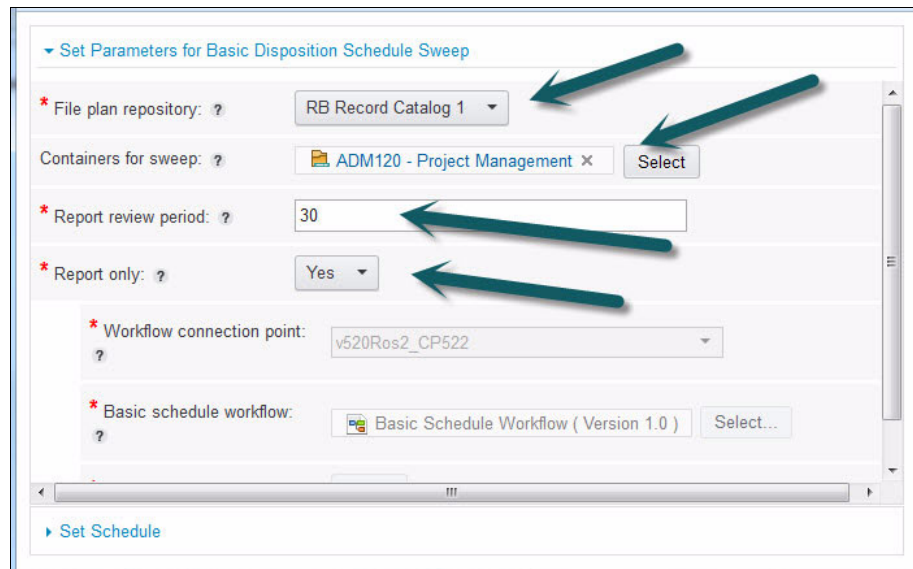


Figure 15-6 Set the parameters for a basic disposition sweep

4. Click **Next** to set the schedule.
5. Enter a name and a description for the sweep. The name is required so that you can identify the sweep after it has completed.
6. Select **Run once** and **Start immediately** as the run options.
7. Click **Schedule Sweep** to schedule the sweep, as shown in Figure 15-7.

Figure 15-7 Setting the sweep schedule and running the sweep

15.2.2 View the sweep results

To view the sweep results, follow these steps:

1. Refresh the tasks lists to verify that the sweep has completed.
2. View the results of the sweep by clicking the **Results** tab, as shown in Figure 15-8 on page 316.

In this example, we see that there are two separate reports generated for each of the record categories that included records ready for destruction because we have configured different values for the Reviewer property. These retention due reports include records that will be ready for destruction 30 days from the date that the sweep was run.

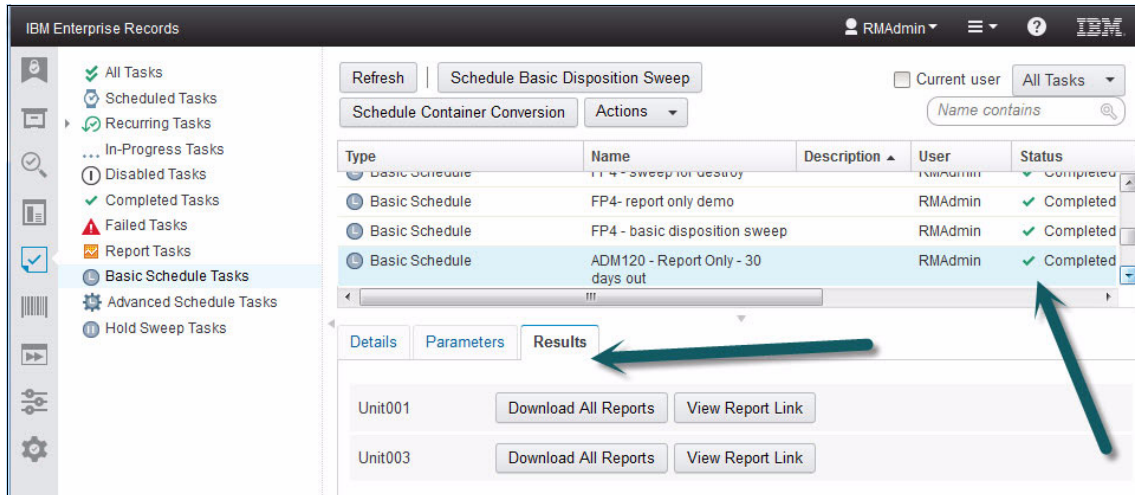


Figure 15-8 View the results of a completed basic disposition sweep configured for report only

15.3 Schedule a basic disposition sweep for immediate destruction

In this section, we demonstrate how to schedule a basic disposition sweep to generate the retention due reports that will be used for immediate destruction of records. We complete the basic disposition process by reviewing the resulting deleted records reports.

15.3.1 Schedule the sweep

To schedule a basic disposition sweep for destruction, follow these steps:

1. Open **Tasks View** from the Enterprise Records desktop, as shown in Figure 15-5 on page 314.
2. From the Schedule menu, select **Schedule Basic Disposition Sweep**.

3. Select the parameters for the basic disposition sweep, as shown in Figure 15-9.

Set Parameters for Basic Disposition Schedule Sweep

* File plan repository: ? RB Record Catalog 1

Containers for sweep: ? ADM120 - Project Management x Select

* Report review period: ? 0

* Report only: ? No

* Workflow connection point: ? v520Ros2_CP522

* Basic schedule workflow: ? Basic Schedule Workflow (Version 1.0) x Select...

* Need approval: ? No

Set Schedule

Figure 15-9 Set the sweep parameters for immediate destruction

In this case, we select the file plan repository, the ADM120 records category for containers for sweep, a report review period of **0** days, set Report Only to **No**, and Set Need Approval to **No**.

4. Then, click **Next** to set the schedule.
5. Enter a name and a description for the sweep. The name is required so that you can identify the sweep after it has completed.

6. Select **Run once** and **Start immediately** as run options, as shown in Figure 15-10.
7. Click **Schedule Sweep** to schedule the sweep. It will start immediately.

The screenshot shows a web-based interface for scheduling a basic disposition sweep. The main panel is titled 'Set Parameters for Basic Disposition Schedule Sweep' and has a sub-section 'Set Schedule'. Under 'Schedule Information', the 'Name' field is filled with 'ADM120 - immediate destruction - due Q4 2014'. The 'Run once' radio button is selected, and the 'Start immediately' checkbox is checked. The 'Schedule Sweep' button is highlighted in the right-hand sidebar. Two green arrows are overlaid on the image: one points to the 'Start immediately' checkbox, and the other points to the 'Schedule Sweep' button.

Figure 15-10 Schedule the sweep for immediate destruction

15.3.2 View the sweep results

The basic disposition sweep results in retention due reports for records that are ready for destruction. Separate workflows are launched to complete the destruction process depending on how the Reviewer property has been set on each record category being processed for basic disposition.

To view the sweep results, follow these steps:

1. Refresh the tasks lists to verify that the sweep has completed.
2. View the results of the sweep by clicking the **Results** tab.

As shown in Figure 15-11, the retention due reports are generated by the sweep and a separate workflow is launched for each reviewer.

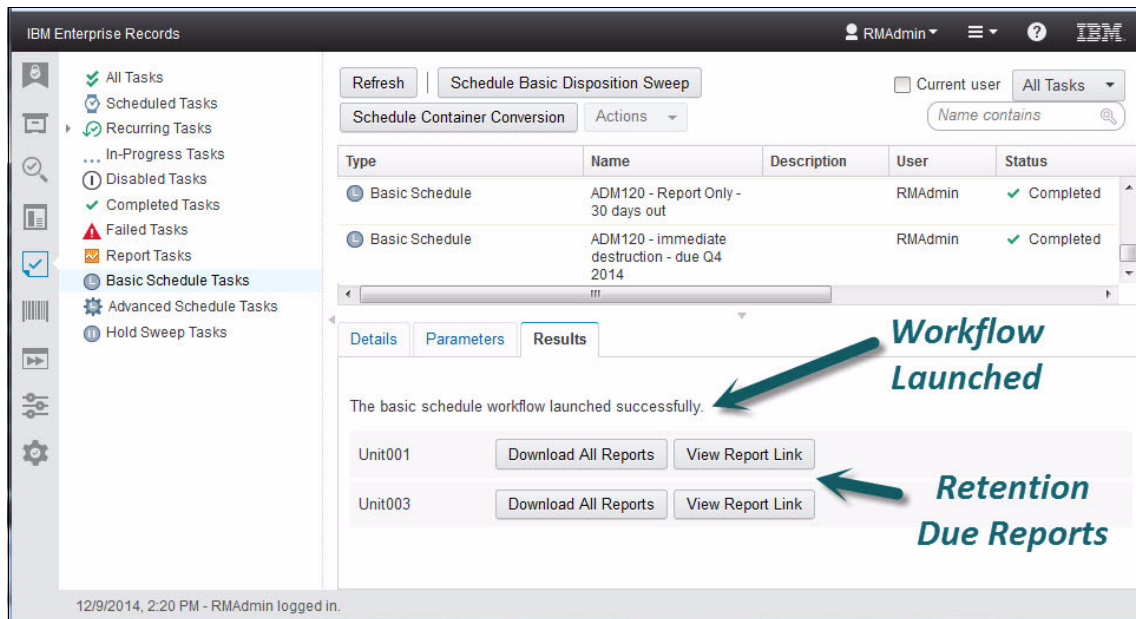


Figure 15-11 View the sweep results to see the retention due reports and verify workflow launched

15.3.3 Verify the destroy results

The results of the destroy workflow would typically be reviewed by a different user than the user who scheduled the sweep.

To verify the destroy results for each workflow, follow these steps:

1. Log in to the Enterprise Records desktop with a user account that has permissions to view and process work.
2. Open **Work View** from the desktop.
3. Expand the process application space for **Records Management**.
4. For Basic Schedule Workflow Reviewer, select **Public Inboxes** to see the work items that are available for review.

Figure 15-12 shows the Work View with the two work items highlighted. These work items have automatically completed the Destroy process and are awaiting Destroy Results review.

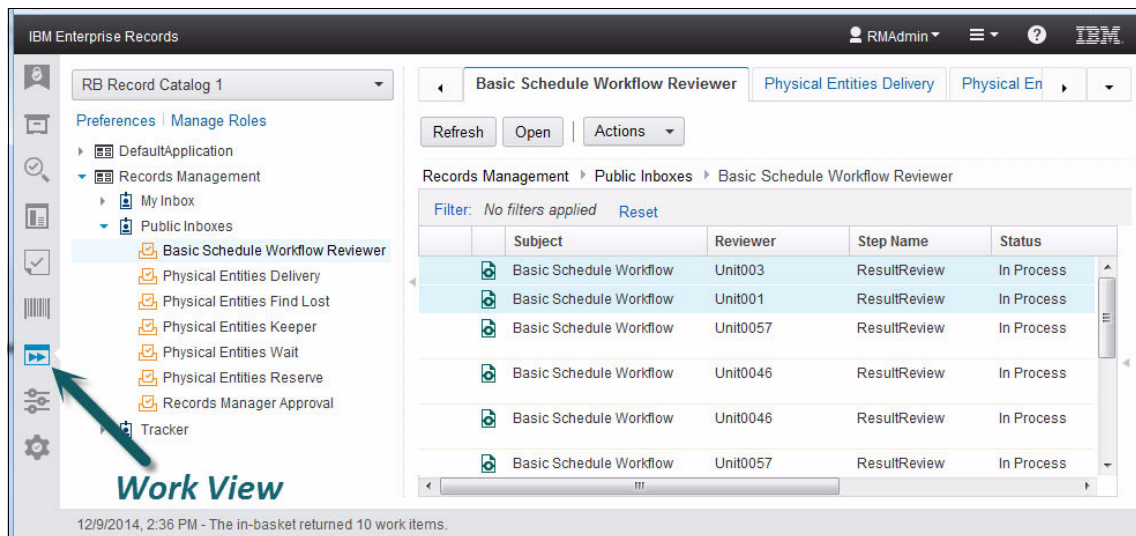


Figure 15-12 Use the work view to display the Basic Schedule Workflow Reviewer inbox

5. Open each work item to view the **ResultReview** step.
6. Click the **Attachments** tab.
7. Click **DestroyResultReports** to see the list of destroy result reports.

As shown in Figure 15-13, the ResultReview step lists the Destroy Result Reports, which include both a Deleted Records report and a Not Deleted Records report. These reports are transcript files that are stored in the FPOS and, optionally, can be declared as records.

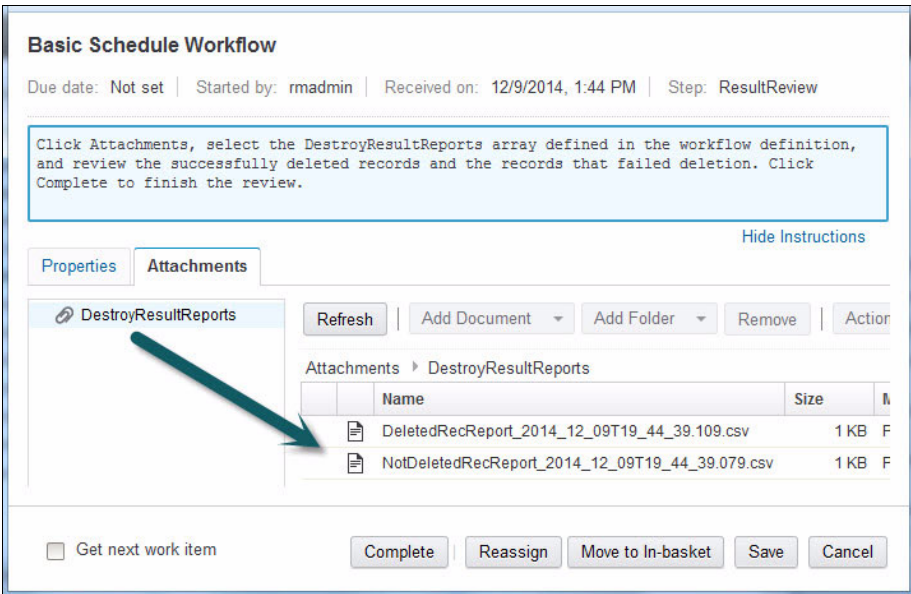


Figure 15-13 ResultReview step shows the destroy results

You can open and view each report if you choose.

8. Click **Complete** to finish the workflow.

15.4 Schedule a basic disposition sweep for approval before destruction

This section explains how to schedule a basic disposition sweep to generate the Retention Due reports that are used for approval before the records are destroyed. We complete the approval step and verify the resulting Destroy reports.

15.4.1 Schedule the sweep

To schedule a basic disposition sweep for approval before destruction, follow these steps:

1. Open **Tasks View** from the Enterprise Records desktop, as shown in Figure 15-5 on page 314.
2. From the Schedule menu, select **Schedule Basic Disposition Sweep**.
3. Select the parameters for the basic disposition sweep to indicate that approval is required before records are destroyed.

Figure 15-14 shows the parameters used in this example to select the file plan repository, the ADM120 record category, set Report Only as **No**, set Report Review Period at **0** days, and set Need Approval as **Yes**.

▼ Set Parameters for Basic Disposition Schedule Sweep

* File plan repository: ? RB Record Catalog 1 ▼

Containers for sweep: ? ADM120 - Project Management × Select

* Report review period: ? 0

* Report only: ? No ▼

* Workflow connection point: ? v520Ros2_CP522 ▼

* Basic schedule workflow: ? Basic Schedule Workflow (Version 1.0) × Select...

* Need approval: ? Yes ▼

► Set Schedule

Figure 15-14 Configure the basic disposition sweep for approval before destruction

4. Click **Next** to set the schedule.
5. Enter a name and a description for the sweep. The name is required so that you can identify the sweep after it has completed.
6. Select **Run once** and **Start immediately** as run options, as shown in Figure 15-10 on page 318.
7. Click **Schedule Sweep** to schedule the sweep. The sweep starts immediately.

15.4.2 View the sweep results

The basic disposition sweep results in retention due reports for records that are ready for destruction. Separate workflows are launched to complete the destruction process depending on how the Reviewer property has been set on each record category being processed for basic disposition.

To view the sweep results, follow these steps:

- 1. Refresh the tasks lists to verify that the sweep has completed.
- 2. View the results of the sweep by clicking the **Results** tab.

Figure 15-15 shows the results of the basic disposition sweep. In this example, there is only one retention due report for one reviewer. The Approval step is part of the Basic Schedule workflow that is launched.

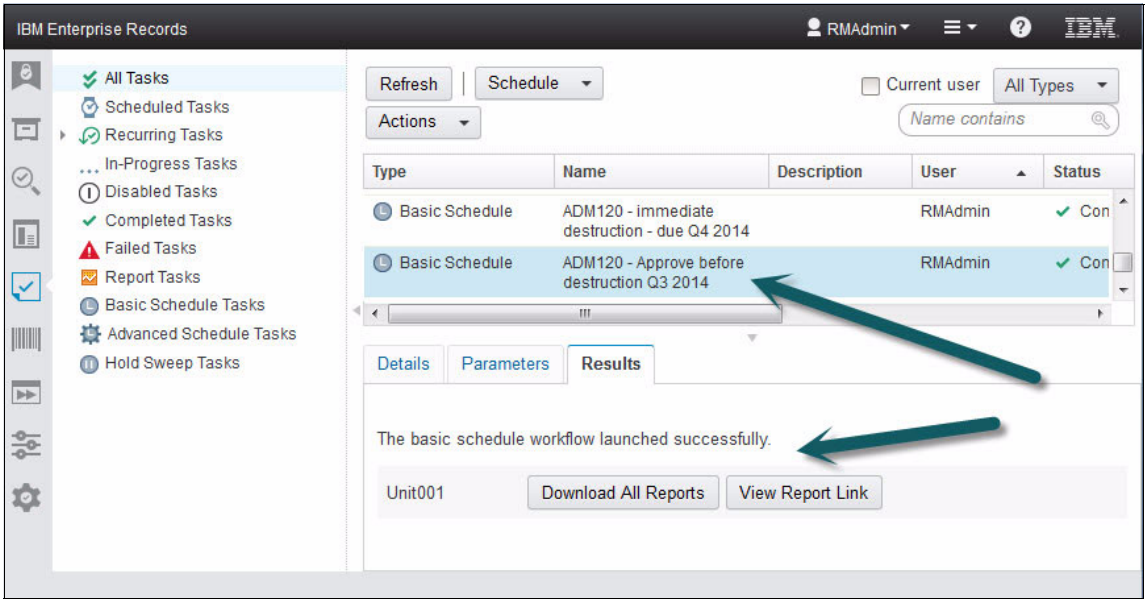


Figure 15-15 Basic disposition sweep results

15.4.3 Approve the records for destruction

When configured for approval, the Basic Schedule workflow includes an approval step that requires user approval before the records in the Retention Due Report are destroyed.

To approve the records for destruction, follow these steps:

1. Log in to the Enterprise Records desktop with a user account that has permissions to view and process work.
2. Open the **Work** view from the desktop.
3. Expand the process application space for **Records Management**.
4. Select **Public Inboxes** for Basic Schedule Workflow Reviewer to see the work items available for review.

Figure 15-16 shows the Work View with one work item awaiting approval. The workflow will not destroy any records until the Approval step has been completed.

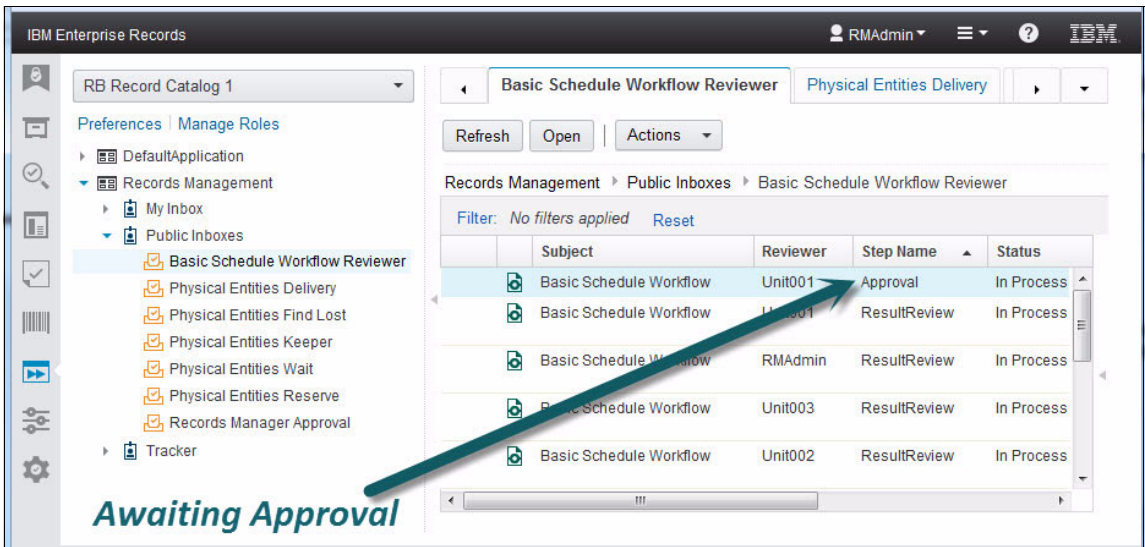


Figure 15-16 Work item awaiting approval before records are destroyed

5. Open the work item to view the retention due report and complete the Approval step.
6. Click the **Attachments** tab.
7. Click the **RetentionDueReport** attachment to see the report listed.

Figure 15-17 shows the Retention Due Report. You can view the details of this report if needed before completing the Approval step.

The Approval steps allow the users who are responsible for the records being destroyed an opportunity to place any records on hold before the system destroys the records. If any of the records listed in the retention due report are placed on hold before the Approval step is completed, those records will not be destroyed upon approval. Only the records that are *not* on hold will be destroyed.

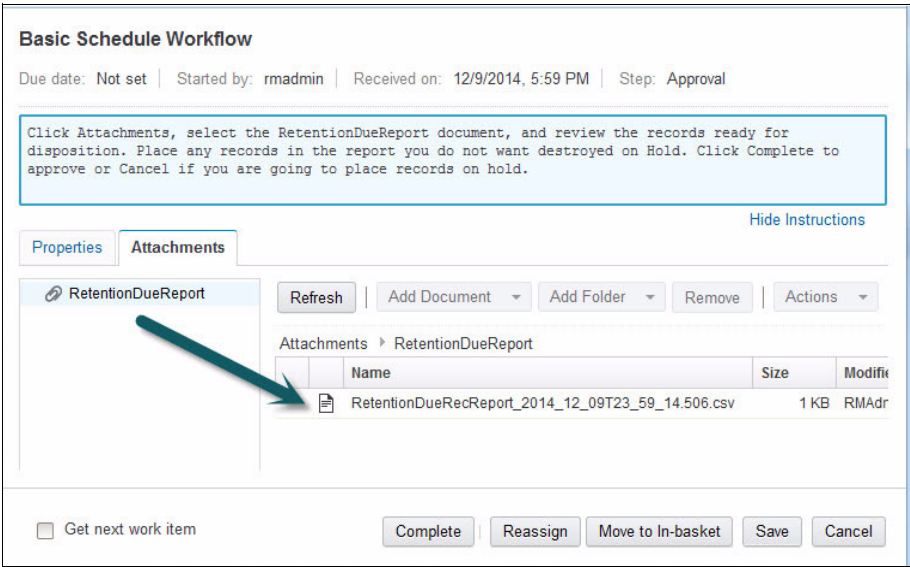


Figure 15-17 Approval step for the basic disposition process

8. Click **Complete** to approve the destruction of the records in the Retention Due Report.

15.4.4 Verify the destroy results

After the Approval step has been completed, the Basic Schedule workflow automatically destroys all records in the retention due report that have not been placed on hold.

To verify the Destroy results, follow these steps:

1. Log in to the Enterprise Records desktop with a user account that has permissions to view and process work if not already logged in.
2. Open **Work View** from the desktop.

3. Expand the process application space for Records Management.
4. Select the **Public Inboxes** for Basic Schedule Workflow Reviewer to see the work items available for review, or click **Refresh** if already displaying the Basic Schedule Workflow Reviewer inbox.
5. Open the work item to view and complete the ResultReview step.
6. Click the **Attachments** tab.
7. Click **DestroyResultReports** to see the list of destruction results (Figure 15-18).

Basic Schedule Workflow

Due date: Not set | Started by: rmdadmin | Received on: 12/10/2014, 9:14 PM | Step: ResultReview

Click Attachments, select the DestroyResultReports array defined in the workflow definition, and review the successfully deleted records and the records that failed deletion. Click Complete to finish the review.

[Hide Instructions](#)

Properties

Attachments

DestroyResultReports

Refresh

Add Document

Add Folder

Remove

Actions

Attachments

DestroyResultReports

	Name	Size	Modified By
	DeletedRecReport_2014_12_11T03_14_23.429.csv	1 KB	P8Admin
	NotDeletedRecReport_2014_12_11T03_14_23.139.csv	1 KB	P8Admin

☐ Get next work item

Complete

Reassign

Move to In-basket

Save

Cancel

Figure 15-18 ResultReview step shows the destroy results after approval

As shown in Figure 15-18, the ResultReview step lists the Destroy Result Reports, which include both a Deleted Records report and a Not Deleted Records report. These reports are transcript files that are stored on the FPOS and can be declared as records (optional).



Advanced disposition case study

This case study describes using advanced disposition schedules and running the advanced disposition sweep.

This chapter explains how to do the following tasks:

- ▶ Configure advanced disposition for approval before destruction
- ▶ Schedule and complete advanced disposition
- ▶ Configure advanced disposition for automatic destruction
- ▶ Schedule and complete advanced disposition for Auto Destroy
- ▶ Convert a record category to a basic schedule

16.1 Configure advanced disposition for approval before destruction

In this section, we demonstrate how to add an advanced disposition schedule to the repository and assign it to the appropriate record category. We configure the advanced disposition schedule to use the Destroy workflow that includes an approval step before destruction.

Advanced disposition schedules are constructed by referencing independent, reusable elements, such as actions and triggers, and then specifying the appropriate retention parameters for each phase.

In this scenario, we construct a disposition schedule for the *0500-030 Agreements* record series in the sample government file plan. This series contains record folders that must be destroyed seven years from when the agreement expired. There is one record folder for each agreement where records related to a specific agreement are declared to the correct agreement folder. We set up the disposition schedule to use folder aggregation so that all of the records in the same record folder will be disposed of together.

16.1.1 Add a Destroy action

To add the Destroy action, follow these steps:

1. Log in to the IBM Enterprise Records desktop as a Records Administrator or Records Manager.
2. Open **Configuration View** and select **Actions** as shown in Figure 16-1 on page 329.
3. Click **Add Action**.
4. Enter the appropriate name and description.
5. Select **Destroy** for the Action Type.
6. Click **Select** to choose the Associated Workflow.
7. Select the version of the Destroy Workflow that you want to use. (Unless you have modified the Destroy Workflow, there will only be one version from which to choose).
8. Click **Add** as shown in Figure 16-2 on page 329 to save the new action.

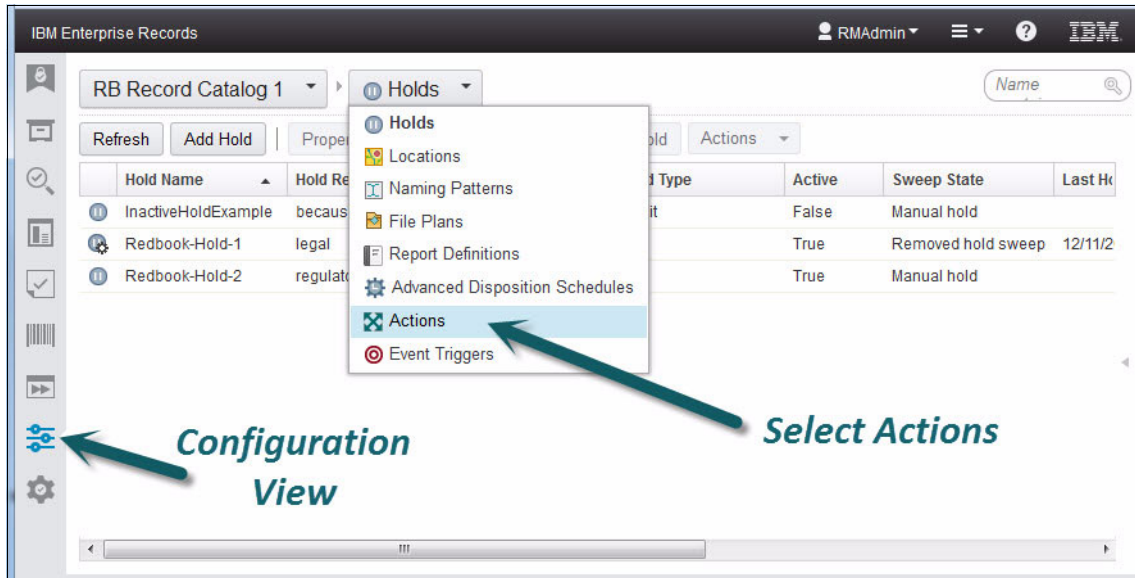


Figure 16-1 Use the configuration view to add and manage actions

The Configuration View is used to configure various elements, such as Actions, Event Trigger, and Holds. In this case, we use the Configuration View to add a new Action.

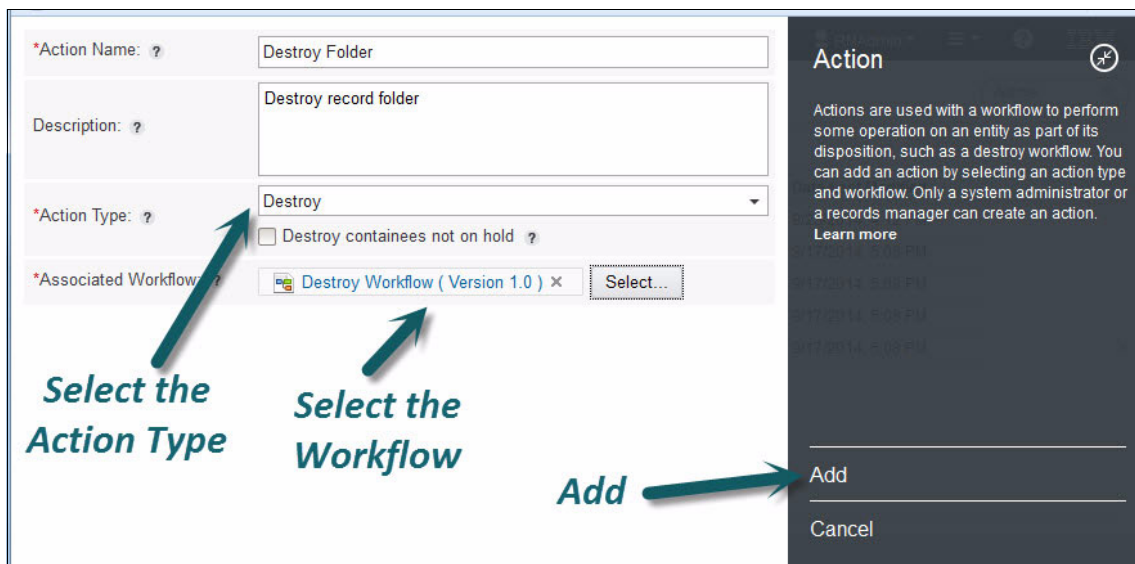


Figure 16-2 The Destroy action type is associated with the Destroy workflow

The Destroy Workflow includes an approval step that must be completed before any records are destroyed.

16.1.2 Add an internal event trigger

To add the internal event trigger, follow these steps:

1. Log in to the Enterprise Records desktop as a Records Administrator or Records Manager.
2. Open **Configuration View** and select **Event Triggers**, as shown in Figure 16-3.

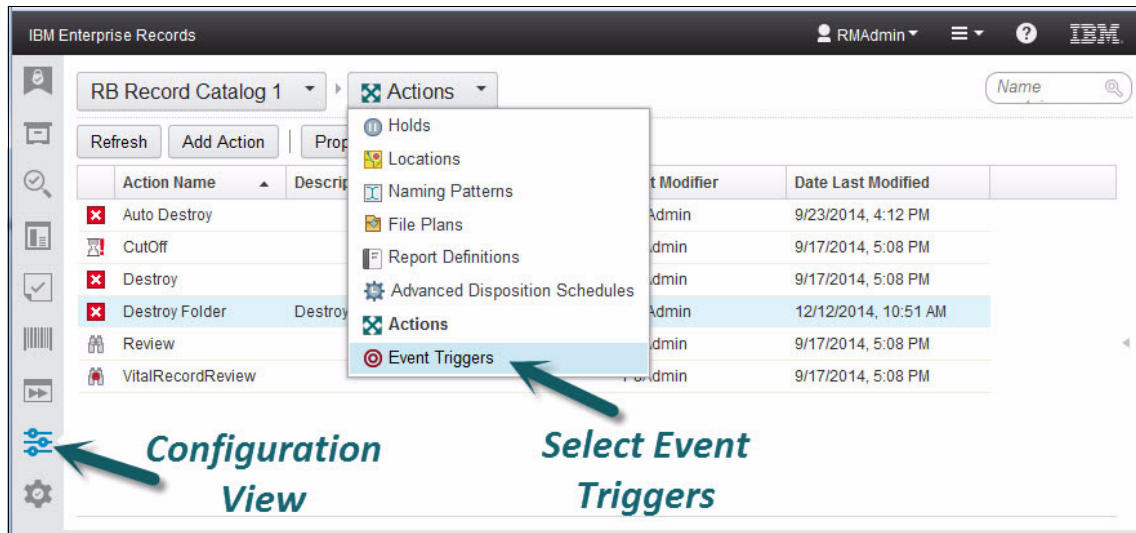


Figure 16-3 Use the configuration view to add and manage vent triggers

3. Click **Add Event Trigger**.

Configure the properties and conditions for the event trigger as shown in Figure 16-4 on page 331.

General

* Event trigger type: ? Internal Event Trigger

Properties

* Internal Event Name: ? Expired (Folder)

Internal Event Description: ? RetentionTriggerDate is not null

* Aggregation: ? Record Folder

Conditions

RetentionTriggerDate ? Is Not Empty

Add Property

☐ Any of the properties ☒ All of the properties

Add Event Trigger

An event is the occurrence of a specified condition based on which the system triggers an action on entities. An event is attached to a disposition schedule and automatically triggers the cut-off for the entity with which the schedule is associated. [Learn more](#)

05/20/2014 11:40 AM

05/20/2014 11:29 AM

05/20/2014 5:02 PM

05/20/2014 2:43 PM

05/20/2014 11:27 AM

Add

Cancel

Figure 16-4 Internal event trigger specifies the aggregation and the trigger conditions

4. Select **Internal Event Trigger** as the event trigger type.
5. Enter an appropriate name and description for the trigger.
6. Select **Record Folder** for the aggregation.
7. Configure the Conditions by selecting the **RetentionTriggerDate** property and the **Is Not Empty** operator.
8. Click **Add** to save the new internal event trigger.

With this configuration, the event trigger applies to record folders with the **RetentionTriggerDate** property.

Note: The *RetentionTriggerDate* property is a custom date property that was added to the default data model as part of the case study object store configuration. This property was added to a valid Record Folder class definition, making it available for use with Enterprise Records.

16.1.3 Add an advanced disposition schedule

After adding the appropriate action and trigger, we now add the disposition schedule that we want to use for the 0500-030 record series in the case study sample file plan for government.

1. Log in to the Enterprise Records desktop as a Records Administrator or Records Manager.
2. Open **Configuration View** and select **Advanced Disposition Schedules**.
3. Click **Add Disposition Schedule**.
4. Enter the name and description for the schedule as shown in Figure 16-5. For this example, we use 0500-030 - Expiration + 7Y for the name.

Figure 16-5 Advanced disposition schedule specifies the trigger, the cutoff, and the phases

- 8. Provide a name for the phase (in this case, we use Destroy) and select **Destroy Folder** for the action. (This is the action we previously configured.)
- 9. Enter 7 for the retention period years.
- 10. Click **Add** to save the advanced disposition schedule.

Figure 16-6 shows the list of advanced disposition schedules for this case study example with the advanced disposition schedule that we just added for record series 0500-030.

Schedule Name	Disposition Authority	Description	Last Modified
0300-130 - Installed + 2Y	AFDA1149	Destroy 2 years after equipment is installed and configured	RMAAdmin
0300-150 - Closed + 7Y	AFDA1155	Destroy 7Y after lease expiration	RMAAdmin
0500-010 - Received + 7Y	AFDA1219	Auto Destroy 7 years after document is received	RMAAdmin
0500-030 - Expiration + 7Y	AFDA1215	Destroy 7 years after expiration or termination	RMAAdmin
0500-130 - Received + 7Y	AFDA1250	Destroy 7 years after the document is received	RMAAdmin
0500-150 - Acquitted + 7Y	AFDA1253	Destroy 7 years after grant acquittal	RMAAdmin
0500-160 - Received + 2Y	AFDA1254	Auto Destroy 2 years after document is received	RMAAdmin
0500-170 - Received + 7Y	AFDA1239	Destroy 7 years after document is received	RMAAdmin

Figure 16-6 List of advanced disposition schedules

After the advanced disposition schedules are added to the configuration, they can be assigned to the appropriate record categories in the file plan.

16.1.4 Assign a disposition schedule to a record category

To assign an advanced disposition schedule to a record category, follow these steps:

- 1. Log in to the Enterprise Records desktop as a Records Administrator or Records Manager.

2. Browse the file plan to locate the record category to which you want to assign the advanced disposition schedule. In this case, we want to find the **0500-030** record category so we can assign the corresponding disposition schedule for that record series.

In Figure 16-7, the record category 0500-030 Agreements, to which we want to assign the disposition schedule, is highlighted.

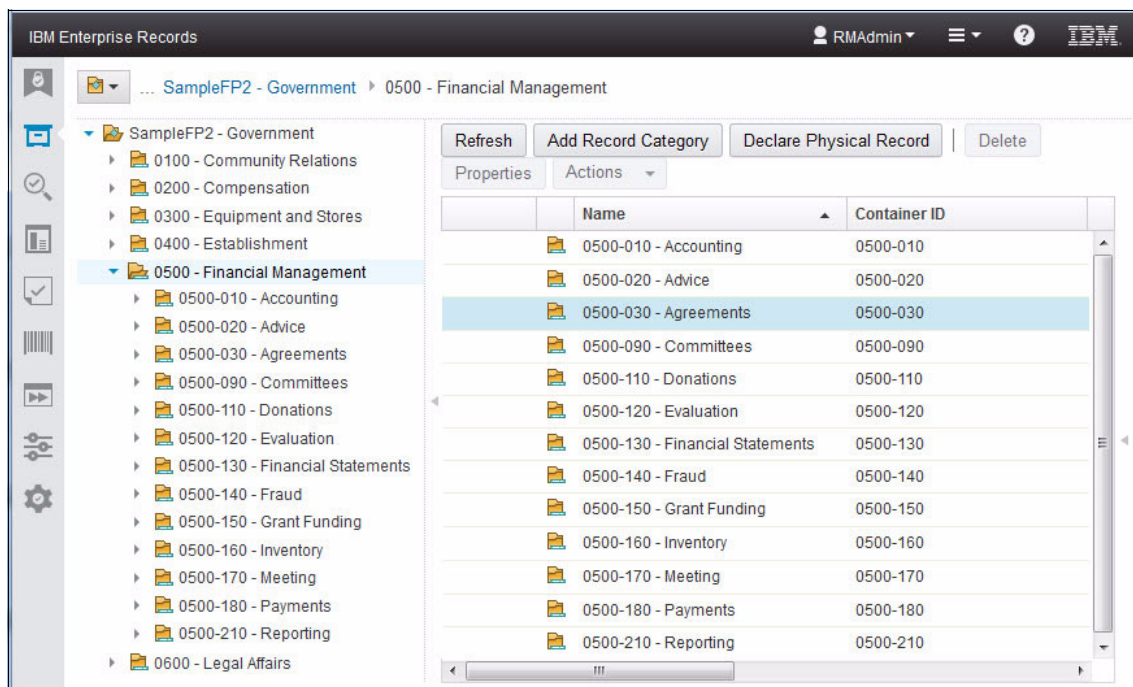


Figure 16-7 Browse to the record category to assign the disposition schedule

3. Click **Properties** to modify the selected record category.

4. Click the **Disposition** tab as shown in Figure 16-8.

The screenshot shows a software interface with a 'Disposition' tab selected. The 'Advanced disposition schedule' radio button is selected. The 'Disposition instructions' field displays '0500-030 - Expiration + 7Y'. The 'Disposition authority' field displays 'AFDA1215'. A green arrow points to the 'Disposition instructions' field with the text 'Disposition schedule is assigned'.

Figure 16-8 Disposition schedule is assigned to the corresponding record category

5. Select **Advanced disposition schedule**.
6. Click **Browse Schedules** to select the correct disposition schedule for this record category. The advanced disposition schedule is displayed in the Disposition Instructions property. Notice that the Disposition Authority, if it has a value, is automatically copied from the advanced disposition schedule to the record category when the schedule is selected.
7. Click **Save** to complete the assignment of the disposition schedule to the record category.

Figure 16-9 shows the advanced disposition schedules assigned to each of the record categories. By running a search for record categories, we can see which schedules are assigned. Using consistent naming conventions for both the record categories in the file plan and the advanced disposition schedules helps in managing the assignment of disposition schedules to record categories.

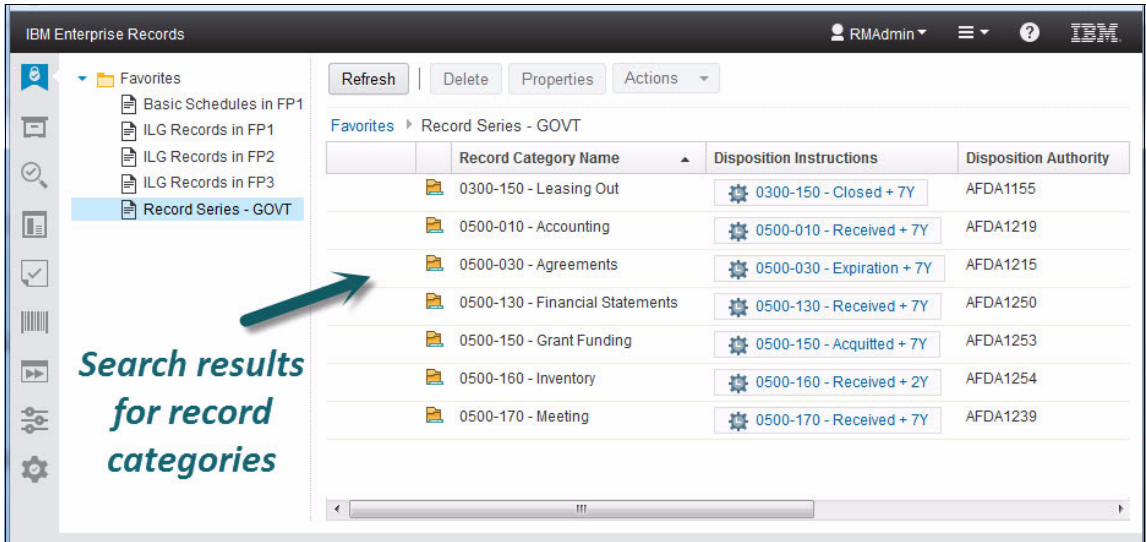


Figure 16-9 Record categories with advanced disposition schedules assigned

16.2 Schedule and complete advanced disposition

In this section, we show how to schedule the advanced disposition sweep to run, how to initiate disposition, and how to complete the advanced disposition process for destruction using our case study example from the government file plan.

16.2.1 Schedule the advanced disposition sweep

In our case study, we assume that some record folders in the 0500-030 Agreements record category will have a trigger date value that will make these folders ready for disposition.

To schedule and run the advanced disposition sweep, follow these steps:

1. Open **Tasks View** from the Enterprise Records desktop.
2. From the Schedule menu, select **Schedule Advanced Disposition Sweep** as shown in Figure 16-10.

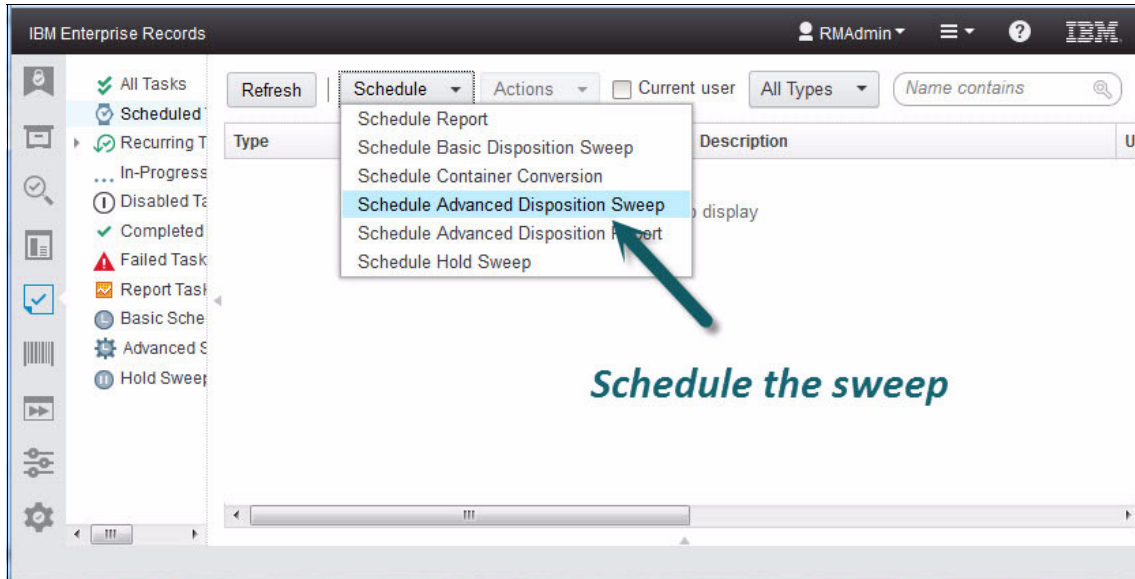


Figure 16-10 Choose to schedule the advanced disposition sweep

3. Select the **File plan repository**, the **Containers for sweep**, and the **Logs output directory** as shown in Figure 16-11.

Set Parameters for Advanced Disposition Sweep

* File plan repository: ? RB Record Catalog 1

* Containers for sweep: ? 0500-030 - Agreements x Select

* Logs output directory: ? VRB Record Catalog 1\Logs

* Type: ? Disposition

* Workflow connection point: ? v520Ros2_CP522

* Run for record types: ? No

* Run for vital: ? No

Set Schedule

Schedule Advanced Disposition Sweep

You can schedule an advanced disposition sweep to initiate retention and disposition of records.

RB Record Catalog 1

RB Record Catalog 1

RB Record Catalog 1

RB Record Catalog 1

Previous

Next

Schedule Sweep

Cancel

Figure 16-11 Set the parameters for advanced disposition sweep

In this case, we select the 0500-030 Agreements records category for containers for sweep to limit the scope of the sweep run.

4. Click **Next** to set the schedule.
5. Enter a name and a description for the sweep. The name is required so that you can identify the sweep after it has completed.

6. Select **Run once** and **Start immediately** as run options, and enter an appropriate user name and password, as shown in Figure 16-12.

Set Parameters for Advanced Disposition Sweep

Set Schedule

Schedule Information

* Name: Sweep 0500-030

Description:

Run once

* Start time: Start immediately

Run on a schedule

* Repeats: Daily

* Start date:

End date:

Login Information

* User name: rmadmin

* Password:

Provide valid credentials

Schedule Advanced Disposition Sweep

You can schedule an advanced disposition sweep to initiate retention and disposition of records.

Previous

Next

Schedule Sweep

Cancel

Figure 16-12 Set the disposition sweep schedule and login information

7. Click **Schedule Sweep** to schedule the sweep. It will start immediately.

16.2.2 Monitor and verify the sweep results

To verify the results of the advanced disposition sweep, follow these steps:

1. **Refresh** the tasks lists to verify that the sweep has completed.

2. View the results of the sweep by clicking the **Results** tab, as shown in Figure 16-13.

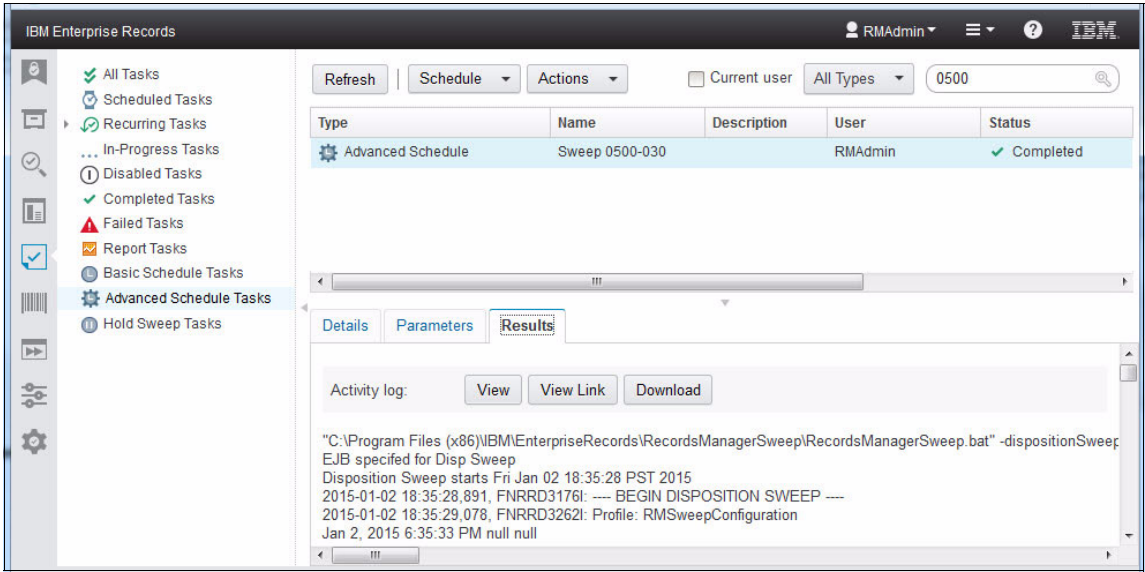


Figure 16-13 Advanced disposition sweep results

The Results tab for the sweep task displays the activity log from the sweep process. You can also verify the sweep results by browsing or searching to see the Ready for Disposition icon next to those items that are ready for processing as shown in Figure 16-14.

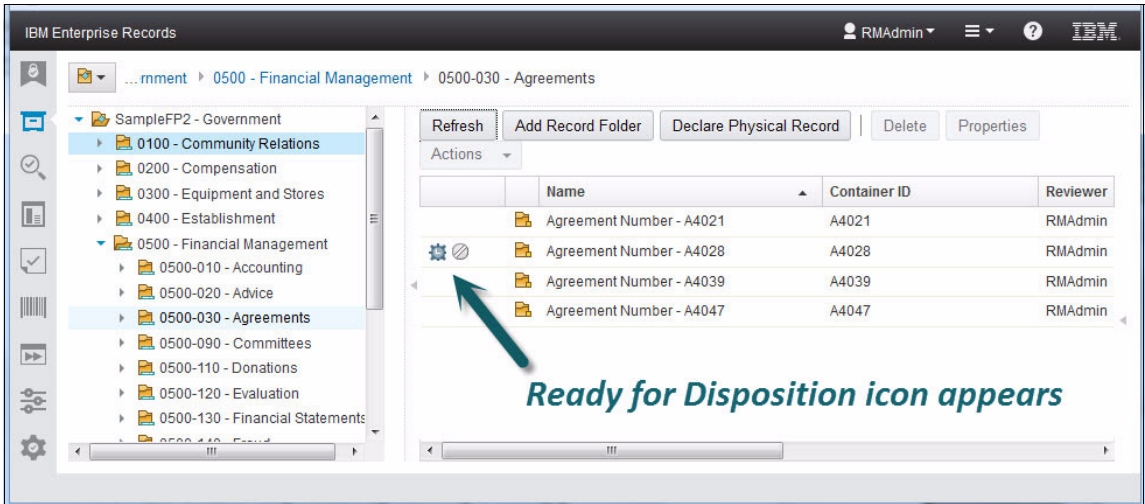


Figure 16-14 Ready for disposition icon appears next to items after running sweep

16.2.3 Initiate disposition by running a sweep

In this section, we show how to initiate disposition by running an advanced disposition sweep where the type of sweep is set for Initiate Disposition.

To initiate disposition by running a sweep, follow these steps:

1. Open **Tasks View** from the Enterprise Records desktop.
2. From the Schedule menu, select **Schedule Advanced Disposition Sweep**.
3. Select the **File plan repository**, the **Containers for sweep**, and the **Logs output directory**, as appropriate.
4. Select **Initiate Disposition** for the type of sweep, as shown in Figure 16-15.

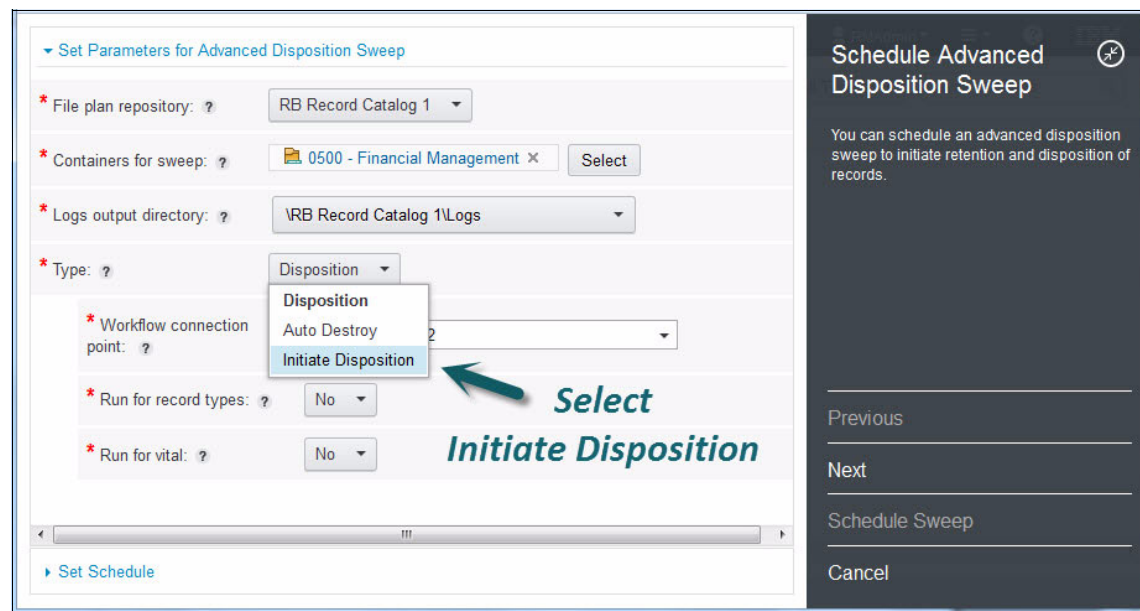


Figure 16-15 Select initiate disposition for the type of sweep

5. Click **Next** to set the schedule.
6. Enter a name and a description for the sweep. The name is required so that you can identify the sweep after it has completed.
7. Select **Run once** and **Start immediately** as run options, and enter an appropriate user name and Ppassword.
8. Click **Schedule Sweep** to schedule the sweep. It will start immediately.

Monitor or verify sweep results (optional)

You can also monitor and verify the sweep results by using the following steps:

1. **Refresh** the tasks lists to verify that the sweep has completed.
2. View the results of the sweep by clicking the **Results** tab as shown in Figure 16-16.

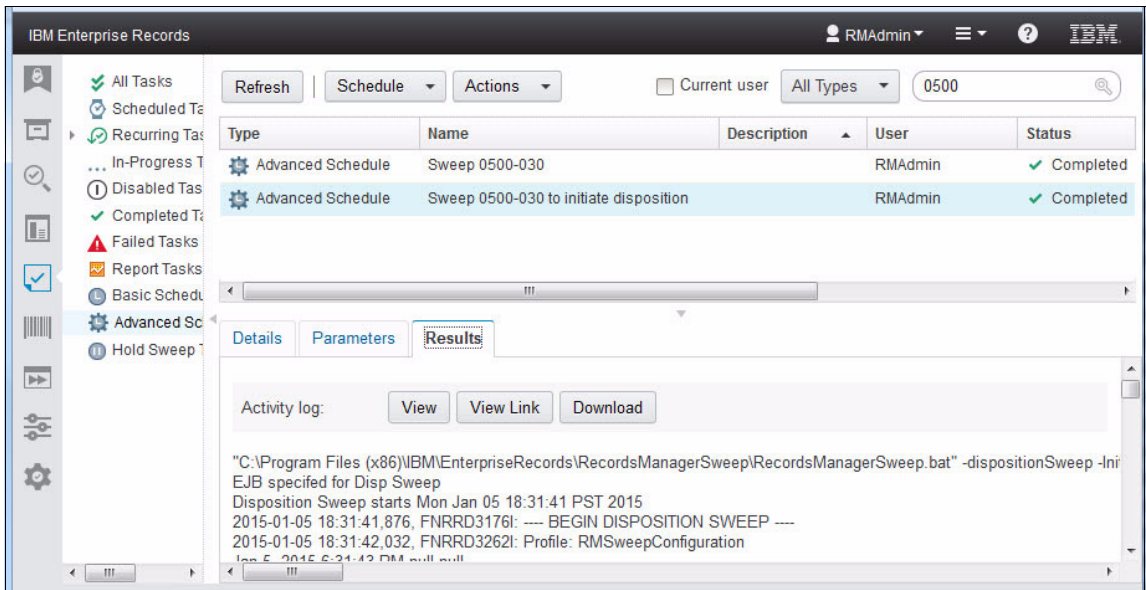


Figure 16-16 Initiate disposition sweep has completed

3. **Browse** the file plan to see that disposition is in progress for the appropriate containers, as shown in Figure 16-17.

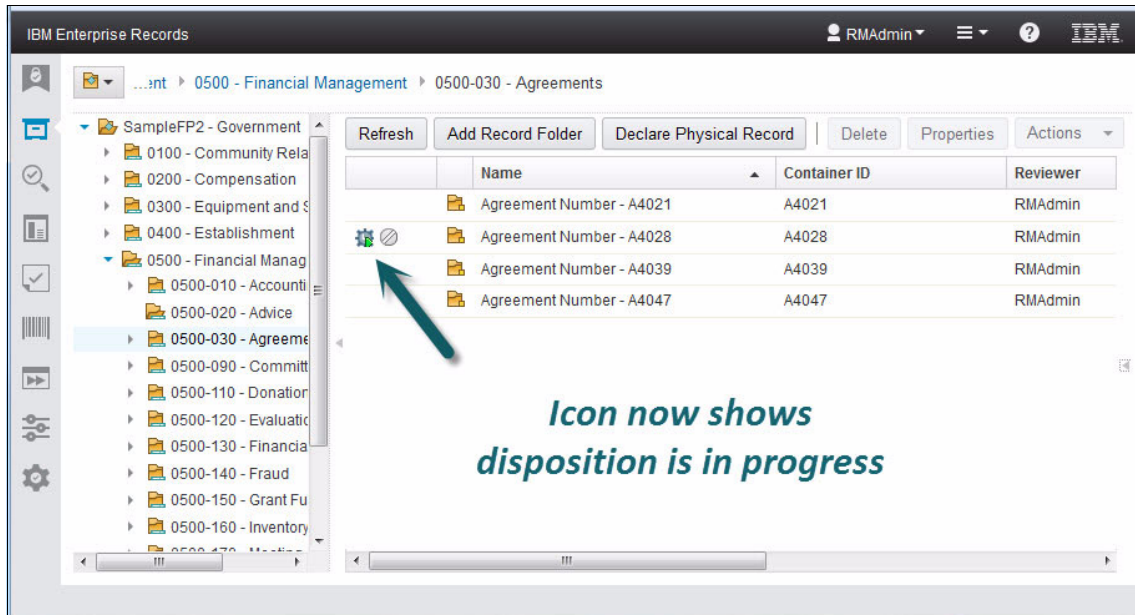


Figure 16-17 After initiating disposition, the icon changes to show that disposition is in progress

16.2.4 Complete the Destroy workflow process

In this section, we show how to complete the disposition process that was initiated in the previous section. In this case study, we have configured the disposition schedule to use the Destroy workflow.

To approve the records for destruction, follow these steps:

1. Log in to the Enterprise Records desktop with a user account that has permissions to view and process work.
2. Open **Work View** from the Enterprise Records desktop.
3. Expand the process application space for **Records Management**.
4. Expand **Public Inboxes**.

5. Select the **Records Manager Approval** queue to see the work items available for review, as shown in Figure 16-18.

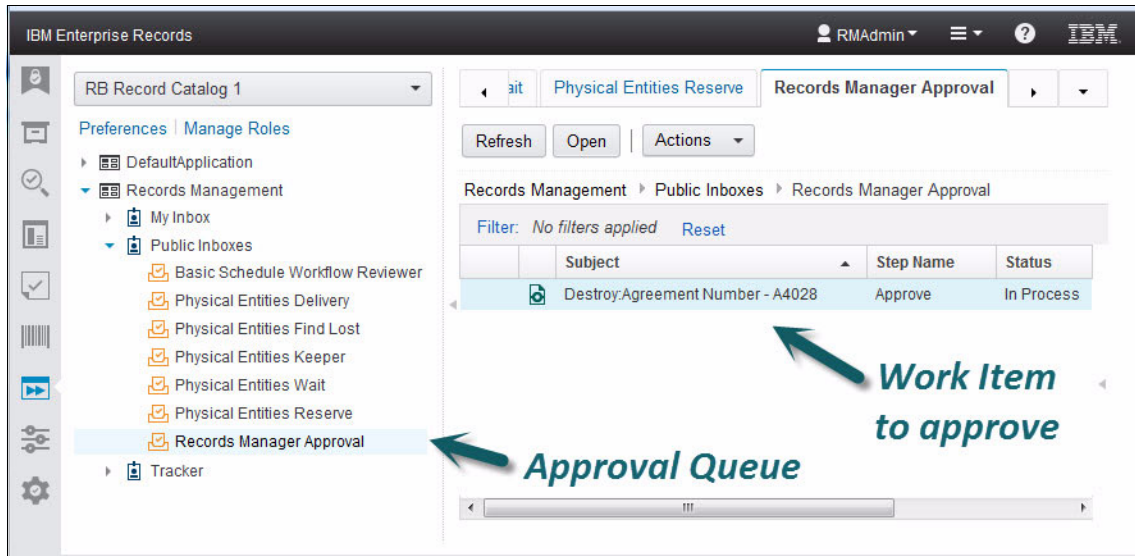


Figure 16-18 Records Manager Approval queue shows work items available to process

In this example, there is one work item to approve for destruction. The subject of the work item indicates *Destroy* along with the name of item to be destroyed, and the step name indicates *Approve* as the action to be completed. In this case, we approve destruction of the record folder that contains all records for Agreement Number - A4028.

6. Open the selected work item to view the Approval step. The list of record folders to approve for destruction is shown in Figure 16-19. In this case, we see only one record folder in the list, the folder for Agreement Number - A4028.

Destroy: Agreement Number - A4028

Due date: Not set | Started by: rmadmin | Received on: 1/5/2015, 6:32 PM | Step: Approve

1. If there is a record container in the list, select that row and choose Review Content from the context menu to review or modify the content of the entity.
2. For the other entity types, select a row and click Properties in the toolbar to review or modify the properties of the entity.

Hide Instructions

Properties History

Refresh Properties Actions

Name	Disposition Instructions	Decision	Comments
Agreement Number - A4028	0500-030 - Expiration + 7Y	Approve Approve Reject	

Approval Decision

☐ Get next work item

Complete Reassign Move to In-basket Save Cancel

Figure 16-19 The Approval step lists items to be approved for destruction

7. By default, **Approve** is selected for the approval decision, as shown in Figure 16-19. You can either Approve or Reject each item listed. In this case, we approve the destruction of the record folder listed. You can enter comments to explain the decision (optional).
8. Click **Complete** to dispatch the work item and continue the Destroy process.
9. **Refresh** the Records Manager Approval queue to review the results of the destruction. After completing the approval step with the decision to approve destruction, all electronic records in the listed record folders that are not on hold will be destroyed.

In this case study, the record folders being destroyed for this record series contain only electronic records, so any additional processing required for physical records will be skipped. A destruction transcript is generated and available for review. Figure 16-20 shows the Transcript step, which is the final step in the Destroy workflow allowing a user to review the destruction transcript.

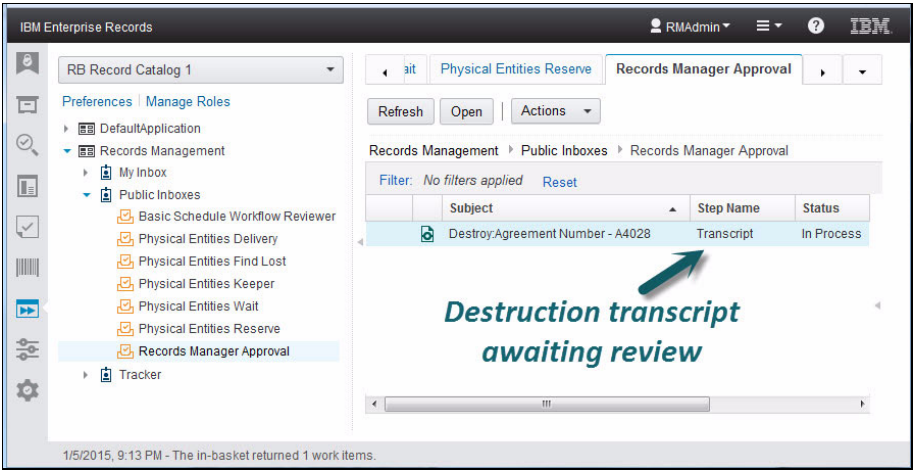


Figure 16-20 The Transcript step allows a user to review the destruction transcript

10. Open the work item to view the Transcript step. You can download and view the destruction transcript (optional). The transcript is an .xml file that is not intended for routine user review.
11. Click **Complete** to dispatch the work item and complete the final step in the Destroy workflow, as shown in Figure 16-21 on page 347.

Destroy: Agreement Number - A4028

Due date: Not set | Started by: rmadmin | Received on: 1/5/2015, 9:12 PM | Step: Transcript

1. In the Transcript file field, locate the transcript file that was created when the Destroy workflow was run.
2. Select the transcript file name to view information about the destroy action for each entity.

[Hide Instructions](#)

Properties

Comment: ?

Transcript file: ? Transcript_Mon Jan 05 21:12:13 PST 2015.xml Actions ▾

☐ Get next work item

Complete Reassign Move to In-basket Save Cancel

Figure 16-21 Complete the transcript step as the final step in the Destroy workflow

16.3 Configure advanced disposition for automatic destruction

Next, we demonstrate how to add an advanced disposition schedule to the repository and assign it to the appropriate record category for automated destruction by using the Auto Destroy feature.

In this scenario, we configure a disposition schedule for the 0500-130 Financial Statements record series in the sample government file plan. This record series is set up for record-level aggregation where records are filed directly in the record series and disposition is handled individually when their retention is time expires.

16.3.1 Add the Auto Destroy action

If an Auto Destroy action has already been added, you can skip these steps. However, if you have a new system, you might need to add the appropriate Auto Destroy action.

To add the Auto Destroy action, follow these steps:

1. Log in to the Enterprise Records desktop as a Records Administrator or Records Manager.
2. Open **Configuration View** and select **Actions** as shown in Figure 16-1 on page 329.
3. Click **Add Action**.
4. Enter the appropriate name and description.
5. Select **Auto Destroy** for the Action Type, as shown in Figure 16-22.

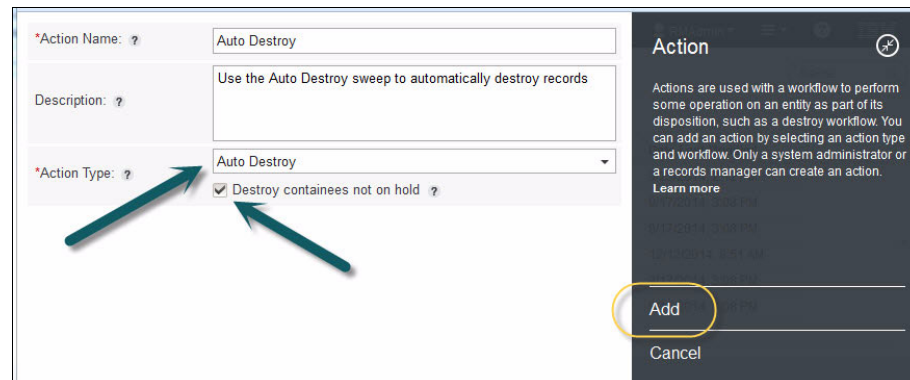


Figure 16-22 Configure an action for Auto Destroy

6. Enable the option to destroy containees not on hold as shown in Figure 16-22.
7. Click **Add** to save the new action.

16.3.2 Add the internal event trigger

To add the internal event trigger, follow these steps:

1. Log in to the Enterprise Records desktop as a Records Administrator or Records Manager.
2. Open **Configuration View** and select **Event Triggers**.
3. Click **Add Event Trigger**.

Configure the properties and conditions for the event trigger as shown in Figure 16-23.

General

* Event trigger type: ? Internal Event Trigger

Properties

* Internal Event Name: ? Received (Record)

Internal Event Description: ? Document Date is not null

* Aggregation: ? Record

Conditions

DocumentDate ? Is Not Empty

Add Property

☐ Any of the properties ☒ All of the properties

Add Event Trigger

An event is the occurrence of a specified condition based on which the system triggers an action on entities. An event is attached to a disposition schedule and automatically triggers the cut-off for the entity with which the schedule is associated. [Learn more](#)

Disposal Trigger

Recalled Trigger

Recalled (Record)

Timeline

Document Date is not null

Record

METADATA EVENT

Add

Cancel

Set the Aggregation

Specify the conditions

Figure 16-23 Internal event trigger configured for record aggregation

4. Select **Internal Event Trigger** as the event trigger type.
5. Enter an appropriate name and description for the trigger.
6. Select **Record** for the aggregation.
7. Configure the **Conditions** by selecting the **DocumentDate** property and the **Is Not Empty** operator.
8. Click **Add** to save the new internal event trigger.

With this configuration, the event trigger applies to records with the DocumentDate property.

Note: The *DocumentDate* property is a custom date property that was added to the default data model as part of the case study object store configuration. This property was added to a valid record class definition, making it available for use with Enterprise Records.

16.3.3 Assign the disposition schedule to the record category

After adding the appropriate action and trigger, we now add the disposition schedule that we want to use for the 0500-030 record series in the case study sample file plan for government.

To add a new disposition schedule, follow these steps:

1. Log in to the Enterprise Records desktop as a Records Administrator or Records Manager.
2. Open **Configuration View** and select **Advanced Disposition Schedules**.
3. Click **Add Disposition Schedule**.
4. Enter the name and description for the schedule as shown in Figure 16-24 on page 351. For this example, we use 0500-130 - Received + 7Y for the name.

In this case, we use a naming convention that refers to the record series to which this schedule will be applied. The name also includes information about the trigger and retention period.

5. Enter information about the disposition authority if appropriate. In this case, we reference an Australian government regulation as an example.
6. Choose **Internal Event Trigger** and select **Received (Record)** for the trigger value. (This is the trigger that we previously configured.)
7. Choose **DocumentDate** for the cutoff base.
8. Provide a name for the phase (in this case we use Auto Destroy) and select **Auto Destroy** for the action. (This is the action we previously configured.)
9. Enter 7 for the retention period years.

▼ Properties

*Schedule Name: ?

0500-130 - Received + 7Y

Description: ?

Destroy 7 years after the document is received

Disposition Authority: ?

AFDA1250

▼ Trigger and Cutoff

Set the condition that triggers a cutoff of entities. Specify when cutoff will occur. By default, the cutoff base date is set to when disposition sweep is run.

*Trigger: ?

Internal Event Trigger ▼

*Trigger value:

Received (Record) x

Select...

Create

*Cutoff base: ?

DocumentDate ▼

Cutoff action: ?

Select...

Create

*Cutoff delay: ?

0 ▴ ▾ Years 0 ▴ ▾ Months 0 ▴ ▾ Days

▼ Phases

Associating phases with a disposition schedule allows different operations on an entity at different intervals.

*Name ?	*Action ?	*Retention period (years, months, days) ?
Auto Destroy	Auto Destroy x	Select...
		7 ▴ ▾ 0 ▴ ▾ 0 ▴ ▾ Options... x

Add Phase

Add Advanced Disposition Schedule

To control retention and disposition of records for advanced users, define and associate an advanced disposition schedule with record containers and record types. In an advanced disposition schedule, you can control event triggers, cutoff action, cutoff offset, phases, and various workflows. [Learn more](#)

- 05/13/2019 PM
- 05/13/2019 PM
- 05/13/2019 PM
- 05/13/2019 PM
- 05/13/2019 PM
- 05/13/2019 PM
- 05/13/2019 PM
- 05/13/2019 PM

Add

Cancel

10. Click **Add** to save the advanced disposition schedule.

16.3.4 Assign the schedule to the correct record category

To assign the advanced disposition schedule to a record category, follow these steps:

1. Log in to the Enterprise Records desktop as a Records Administrator or Records Manager.
2. **Browse** the file plan to locate the record category to which you want to assign the advanced disposition schedule. In this case, we want to find the **0500-130** record category so we can assign the corresponding disposition schedule for that record series.
3. Click **Properties** to modify the selected record category.
4. Click the **Disposition** tab.
5. Select **Advanced disposition schedule**.

6. Click **Browse Schedules** to select the correct disposition schedule for this record category. The advanced disposition schedule is displayed in the Disposition Instructions property as shown in Figure 16-25.

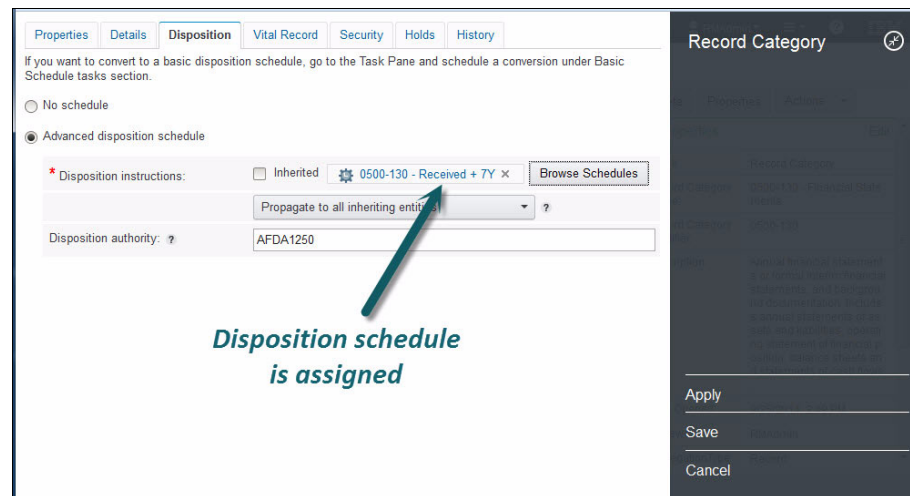


Figure 16-25 Disposition schedule is assigned to the corresponding record category

7. Click **Save** to complete the assignment of the disposition schedule to the record category.

16.4 Schedule and complete advanced disposition for Auto Destroy

In this section, we process the 0500-130 Financial Statements record category for disposition to illustrate advanced disposition using Auto Destroy with record level aggregation.

16.4.1 Schedule the advanced disposition sweep

To schedule the advanced disposition sweep, follow the same steps listed in 16.2.1, “Schedule the advanced disposition sweep” on page 336, but ensure that the appropriate record category (in this example, 0500-130 Financial Statements) is selected for the containers to sweep, as shown in Figure 16-26 on page 353.

16.4.2 Schedule Auto Destroy

Auto Destroy should only be scheduled to run after the advanced disposition sweep has successfully completed.

To schedule Auto Destroy, follow these steps:

1. Open **Tasks View** from the Enterprise Records desktop.
2. From the Schedule menu, select **Schedule Advanced Disposition Sweep**.
3. Select the File plan repository, the Containers for sweep, and the Logs output directory as appropriate.
4. Select **Auto Destroy** for the type of sweep, as shown in Figure 16-28.

▼ Set Parameters for Advanced Disposition Sweep

* File plan repository: ? RB Record Catalog 1

* Containers for sweep: ? 0500-130 - Financial Statements x Select

* Logs output directory: ? \RB Record Catalog 1\Logs

* Type: ? Auto Destroy

Generate transcript: ? Yes

Select Auto Destroy

▶ Set Schedule

Schedule Advanced Disposition Sweep

You can schedule an advanced disposition sweep to initiate retention and disposition of records.

Auto Destroy	Complete
Auto Destroy	Complete
Auto Destroy	Complete
Auto Destroy	Complete
Auto Destroy	Complete
Auto Destroy	Complete
Previous	Complete
Next	Complete
Schedule Sweep	Complete
Cancel	

Figure 16-28 Schedule the sweep to run Auto Destroy

5. Click **Next** to set the schedule.
6. Enter a name and a description for the sweep. The name is required so that you can identify the sweep after it has completed.
7. Select **Run once** and **Start immediately** as run options, and enter an appropriate user name and password.
8. Click **Schedule Sweep** to schedule the sweep. It will start immediately.

Monitor and verify the sweep results (optional)

You can monitor and verify the sweep results by using the following steps:

1. **Refresh** the tasks lists to verify that the sweep has completed.
2. View the results of the sweep by clicking the **Results** tab, as shown in Figure 16-29.

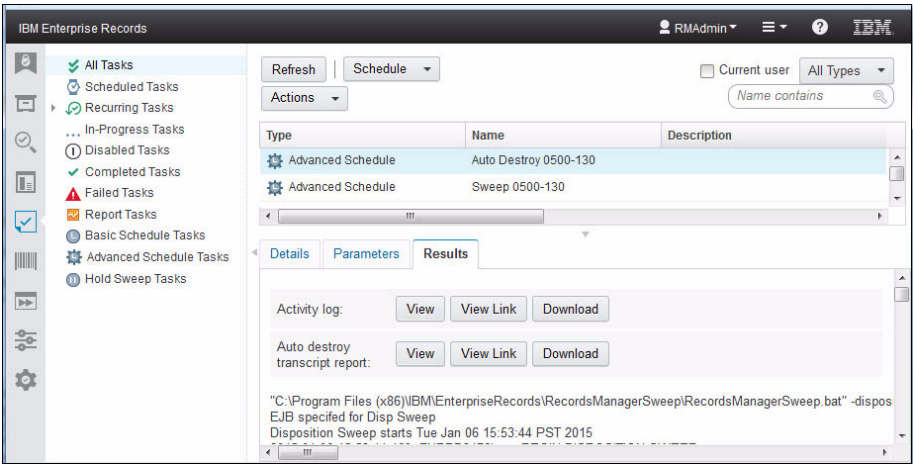


Figure 16-29 Results after Auto Destroy has completed

You can choose to view or download either the activity log or the Auto Destroy transcript report.

After Auto Destroy has successfully completed, the records have been destroyed without any approval or review.

Figure 16-30 shows the 0500-130 Financial Statements record category after Auto Destroy has completed. The records that were previously ready for disposition are now gone.

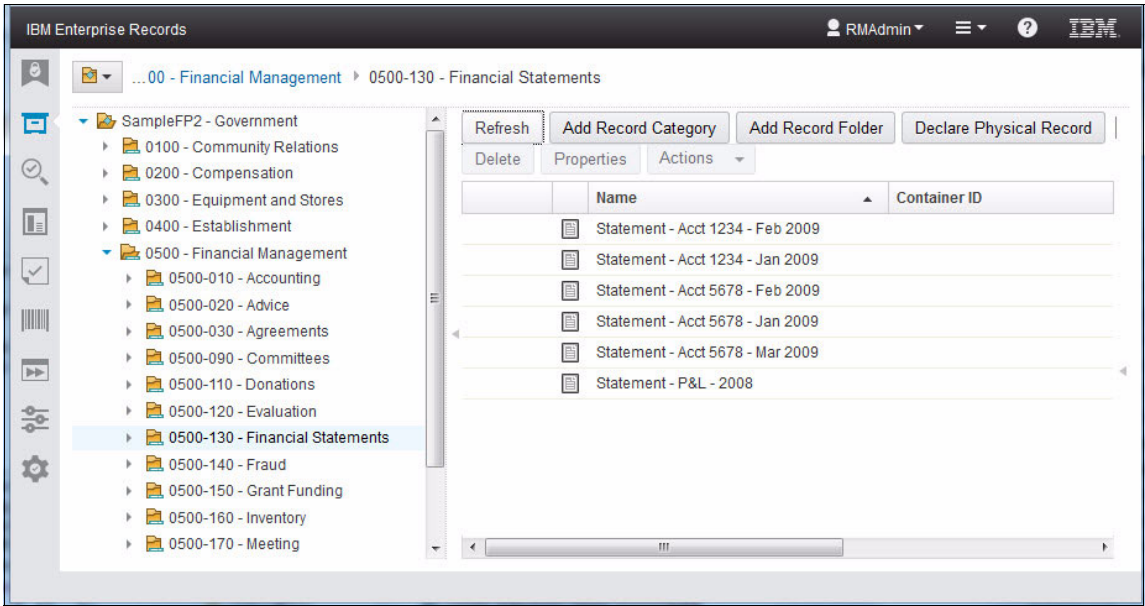


Figure 16-30 After auto destroy completes, records destroyed no longer appear in the record category

16.5 Convert a record category to a basic schedule

In this section, we demonstrate how to convert a record category that has an advanced schedule to use a basic disposition schedule.

16.5.1 Identify an eligible record category

In our case study, the 0500-130 Financial Statements record category is currently configured for advanced disposition. We now convert this record category to have a basic disposition schedule.

Figure 16-31 shows the current configuration for 0500-130 Financial Services. The disposition instructions column shows the advance disposition schedule associated with this record category.

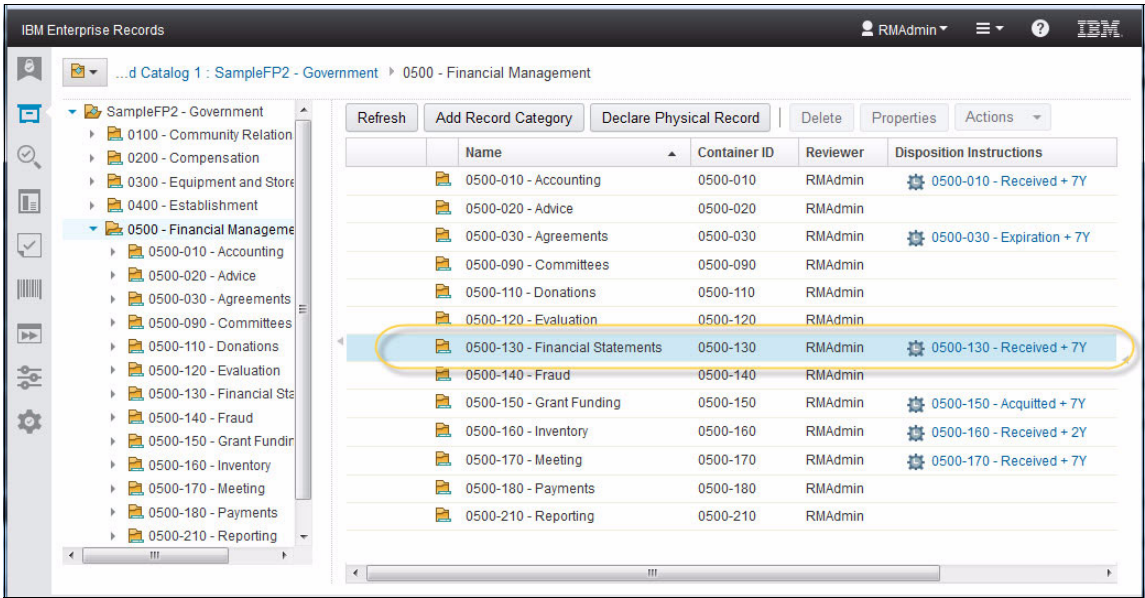


Figure 16-31 Record category with an advanced disposition schedule can be converted

16.5.2 Schedule the container conversion

To schedule the record category for conversion, follow these steps:

1. Open **Tasks View** from the Enterprise Records desktop.
2. From the Schedule menu, select **Schedule Container Conversion**, as shown in Figure 16-32 on page 358.

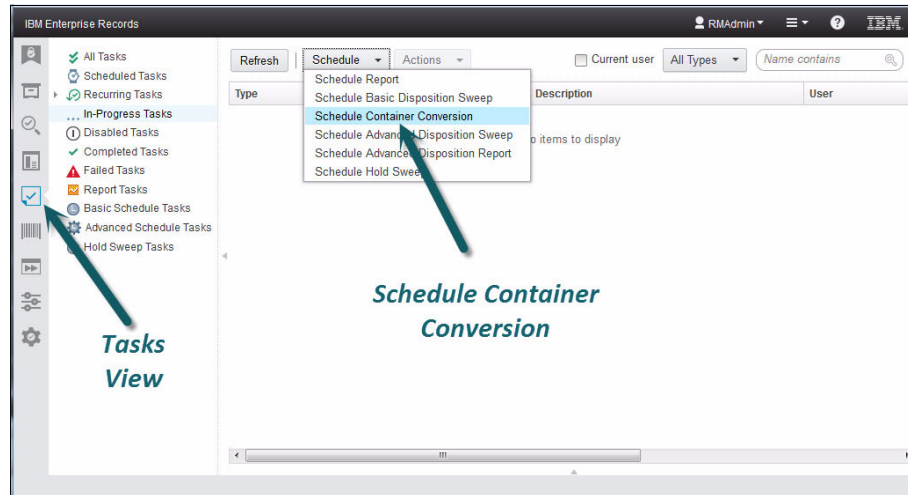


Figure 16-32 Schedule container conversion from the Tasks view

3. Select the record category to convert, as shown in figure Figure 16-33. For this case study, we select **0500-130 Financial Services**.

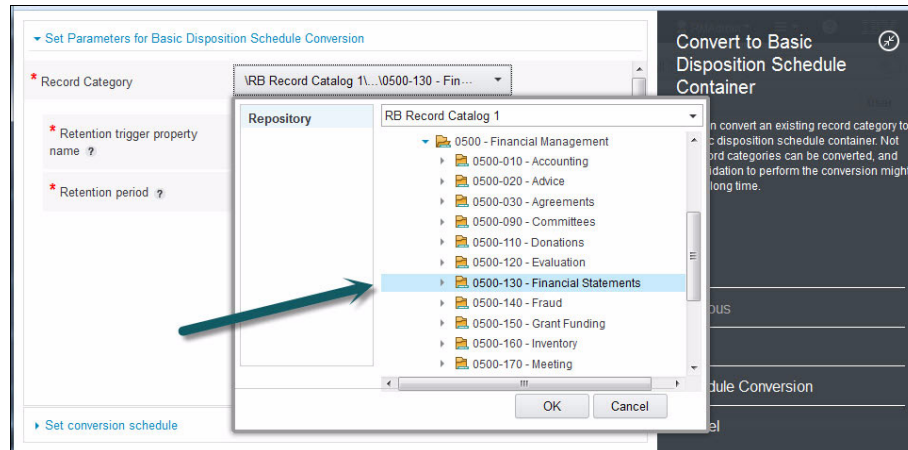


Figure 16-33 Select a record category for conversion

4. Select the correct Retention trigger property, in this case **DocumentDate**.

- Set the correct Retention period, in this case, 7 years, as shown in Figure 16-34.

Set Parameters for Basic Disposition Schedule Conversion

Record Category: IRB Record Catalog 1\...\0500-130 - Fin...

Retention trigger property name: DocumentDate

Retention period: 7 Years 0 Months 0 Days

Set conversion schedule

Convert to Basic Disposition Schedule Container

You can convert an existing record category to a basic disposition schedule container. Not all record categories can be converted, and the validation to perform the conversion might take a long time.

Previous

Next

Schedule Conversion

Cancel

Figure 16-34 Set the retention parameters

- Click **Next** to view the conversion schedule, as shown in Figure 16-35.

Set Parameters for Basic Disposition Schedule Conversion

Set conversion schedule

Schedule Information

Name: Convert container: 0500-130 - Financial Statements

Description: Converting 0500-130 - Financial Statements to a basic disposition schedule container on 4:56 PM.

Run once

Start time: immediately

Login Information

User name

Convert to Basic Disposition Schedule Container

You can convert an existing record category to a basic disposition schedule container. Not all record categories can be converted, and the validation to perform the conversion might take a long time.

Previous

Next

Schedule Conversion

Cancel

Figure 16-35 Set the conversion schedule information

- Click **Schedule Conversion** to start the task.
- Click **Refresh** to monitor the task for completion.

Figure 16-36 shows the completed conversion task.

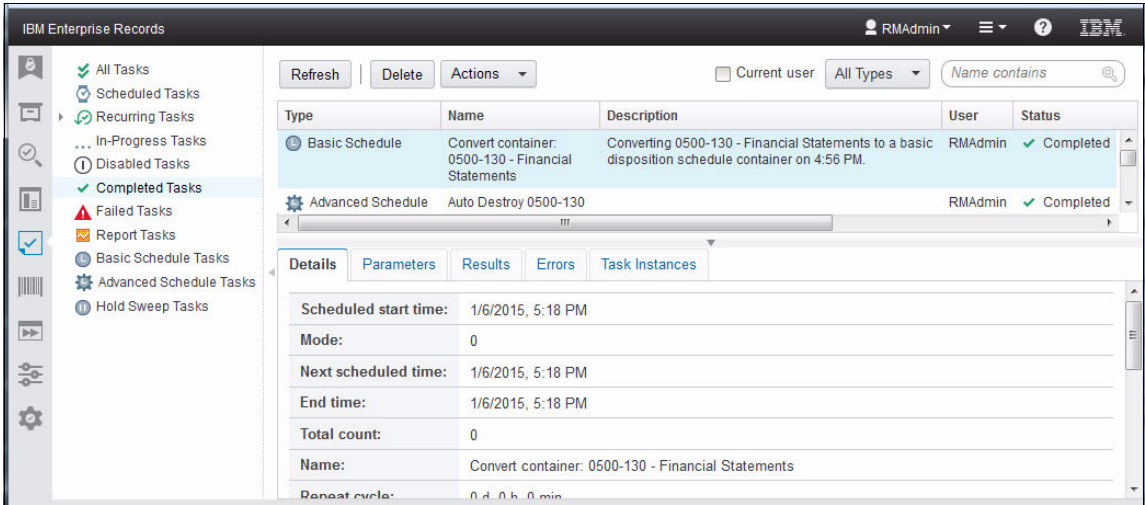


Figure 16-36 Container conversion has completed

(Optional) You can **Browse** to see the updated basic schedule configuration, as shown in Figure 16-37. The disposition instructions for the 0500-130 Financial Services record category display the *basic* disposition schedule configuration. This record category can no longer use advanced disposition.

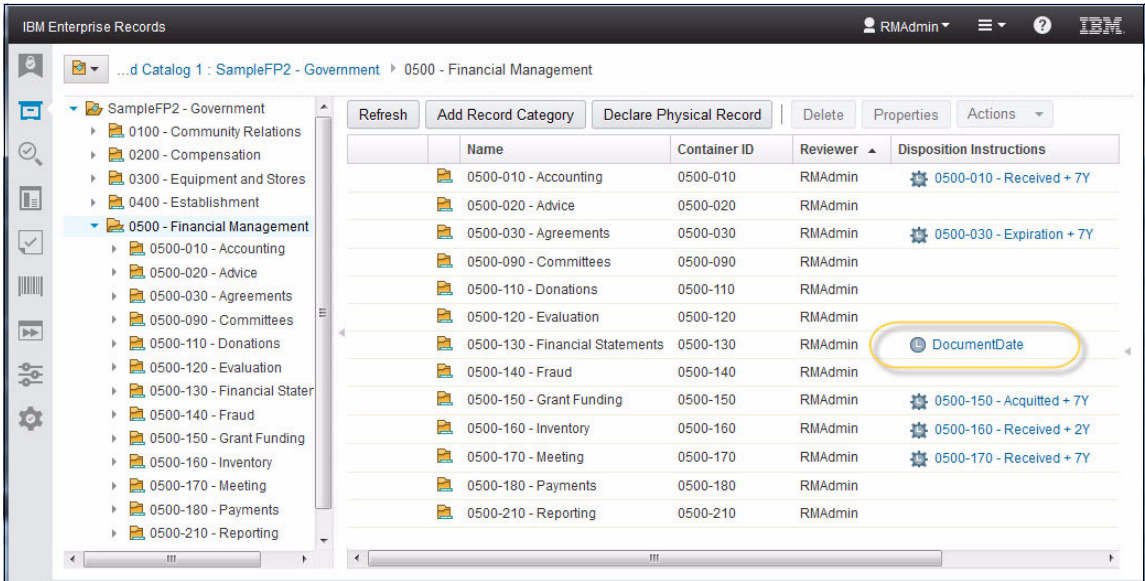


Figure 16-37 Disposition instructions column shows the basic schedule configuration



Records hold case study

In Chapter 8, “Holds and preservation” on page 213, we described the concept of *records hold*. In this chapter, we show you the steps required to perform hold-related activities.

We cover the following topics:

- ▶ Case study hold scenarios
- ▶ Creating a hold
- ▶ Manually placing and removing holds
- ▶ Dynamic holds and Hold Sweep

17.1 Case study hold scenarios

this case study is based on the loan department of the fictional BNK100 bank. Figure 17-1 illustrates the part of the file plan that shows categories that are related to the case study for this book.

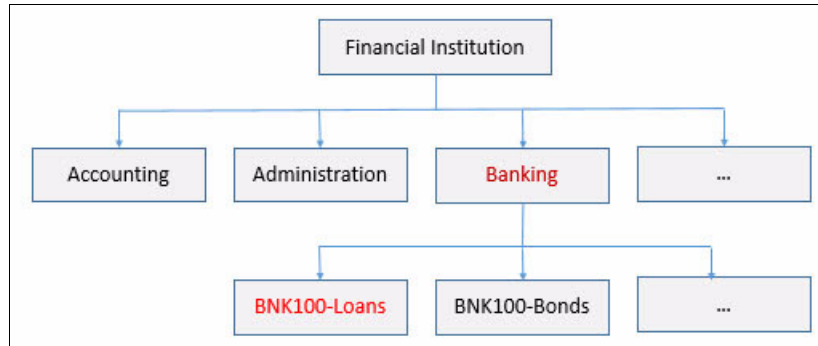


Figure 17-1 Partial file plan showing categories related to the case study for this book

Depending on litigation or business practices, your company might be asked to put specific records on hold. For this case study, we created scenarios for when records might be put on hold:

- Lawsuit about loan 1234

There are disputes associated with particular loan cases. The company is required to put all records associated with a certain loan number on hold.

Case study hold condition: Loan Number = 1234

- Investigation of financial practices

Users might want to browse through the company's file plan or search 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.

17.2 Creating a hold

In this section, we show you how to create a hold, using the scenario described in the previous section, the lawsuit about loan 1234. The company is required to put all records related to that loan on hold. The case study hold condition is Loan Number = 1234.

The hold is uses two condition criteria:

- ▶ KeyValueDescription = Loan Number
- ▶ KeyValue = 1234

Note: Always carefully consider the criteria that you use for dynamic holds so that there is no large number of entities being placed on hold unintentionally.

To create a hold, follow these steps:

1. Launch IBM Enterprise Records web application. Log in as a Records Manager or Records Administrator.
2. Select the **Open Configuration View** icon, and **Hold** from the drop-down menu (Figure 17-2).

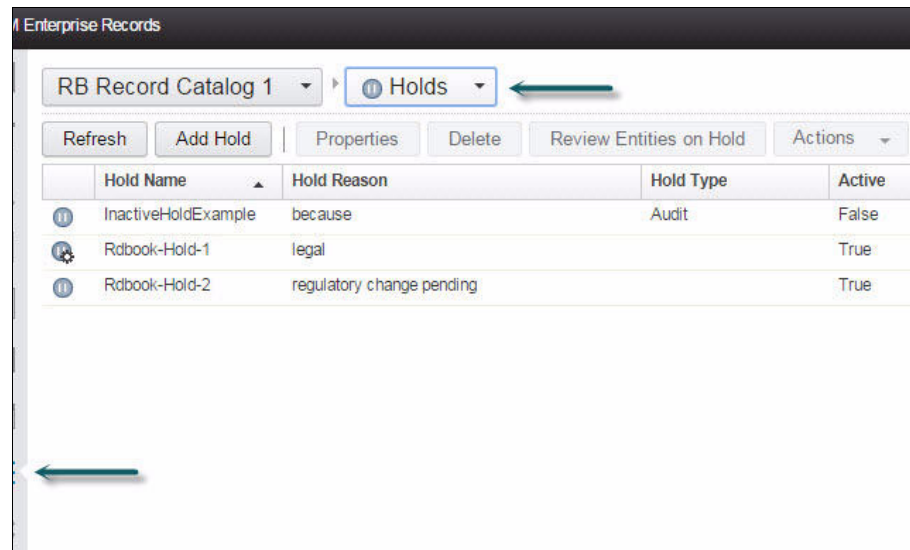


Figure 17-2 Hold configuration view

- d. Click **Add Hold**.
3. Set the hold properties:
 - Enter a name for the hold.
For this case study, we enter Lawsuit On Loan 1234.
 - Specify a reason for adding this hold.
For this case study, we enter RdBook case study.
 - Select the type of hold.
For this case study, we select **Litigation** from the drop-down menu.

- 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.

4. Expand **Condition** to proceed to set the hold conditions.

Alternatively, you can click **Add** (or **Cancel**). If you finish without setting any conditions, this hold can be placed only manually. Hold Sweep will ignore it.

5. Set the hold conditions.

You can set conditions for records, categories, record 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 the entities that meet the criteria on hold.

To set the condition:

- a. Because we are putting records on hold in this case study, select the **Record** tab (it is selected by default).
 - b. Set the first condition by selecting the **KeyValueDescription** property, the operator **Equals**, and the value **Loan Number**.
 - c. Click **Add Property** to add a new line condition.
 - d. Set the second condition by selecting the property **KeyValue**, the operator **Equals** and the value **1234**.
 - e. Select **All of the properties** to create the condition with an AND operator. This ensures that only the records that match both criteria simultaneously are returned. Selecting **Any of the properties** would return any record for which the KeyValueDescription is about a Loan Number but not a particular one and all records with a KeyValue of 1234, even though they are not related to a loan.
6. Verify that the query returns the expected records by clicking **Preview**.

This selection returns all records related to the loan number 1234, as shown in Figure 17-3 on page 367.

▼ Properties

*Hold Name: ?

Lawsuit on loan 1234

Hold Reason: ?

RdBook case study

Hold Type: ?

Litigation

Active: ?

True

▼ Conditions

Record (0)

Record Category (0)

Record Folder (0)

Record Volume (0)

KeyValueDescription ▼ ?

Equals ▼

Loan Number

KeyValue ▼ ?

Equals ▼

1234

Add Property

☐ Any of the properties

☒ All of the properties

Content Contains

☐ Any of the terms

☒ All of the terms

Preview

▼ Preview Results

	Document Title	Last Modifier ▲	Date Last Modified	KeyValueDescription	KeyValue
	LoanNumber-1234-Amendment-July-2014.pdf	RMAdmin	1/14/2015, 9:21 AM	Loan Number	1234
	LoanNumber-1234-SignaturePage.pdf	RMAdmin	1/14/2015, 9:14 AM	Loan Number	1234

Figure 17-3 Create the dynamic hold to put all records related to loan 1234 on hold

- Click **Add** to save the new hold.

The Holds list now shows the newly added hold, as show in Figure 17-4 on page 368.

RB Record Catalog 1 ▾					
Holds ▾					
<div>Refresh</div> <div>Add Hold</div> <div>Properties</div> <div>Delete</div> <div>Review Entities on Hold</div> <div>Actions ▾</div>					
	Hold Name ▴	Hold Reason	Hold Type	Active	Sweep State
ⓘ	InactiveHoldExample	because	Audit	False	Manual hold
ⓘ	Investigation of Loan Practices	RdBook case study	Audit	True	Manual hold
⚙️	Lawsuit on loan 1234	RdBook case study	Litigation	True	Active hold sweep: pending
⚙️	Rdbook-Hold-1	legal		True	Removed hold sweep
ⓘ	Rdbook-Hold-2	regulatory change pending		True	Manual hold

Figure 17-4 Lawsuit on loan 1234: Created

Because the hold was set with conditions, it qualifies as *dynamic hold*. The Sweep State of this hold is Active Hold Sweep: Pending, which means that Hold Sweep has not been run yet.

17.3 Manually placing and removing holds

You can manually put entities on hold and manually remove them.

To place entities on hold automatically, or *dynamically*, use Hold Sweep. See 17.4, “Dynamic holds and Hold Sweep” on page 373.

17.3.1 Manually placing an entity on hold

To manually place a hold, follow these steps:

1. From the Enterprise Records web application, navigate to the entities that you want to manually place on hold.

For this case study, we created several loan documents by navigating to **SampleFP1** → **BNK - Banking** → **BNK100 - Loans** category (Figure 17-5).

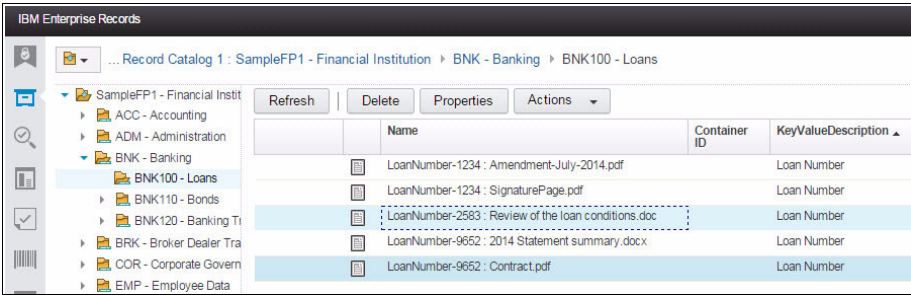


Figure 17-5 Navigating to the list of loans documents

2. Select one or more records that you want to put on hold, right-click, and select the **Place on Hold** action from the pop-up menu, as shown in Figure 17-6.

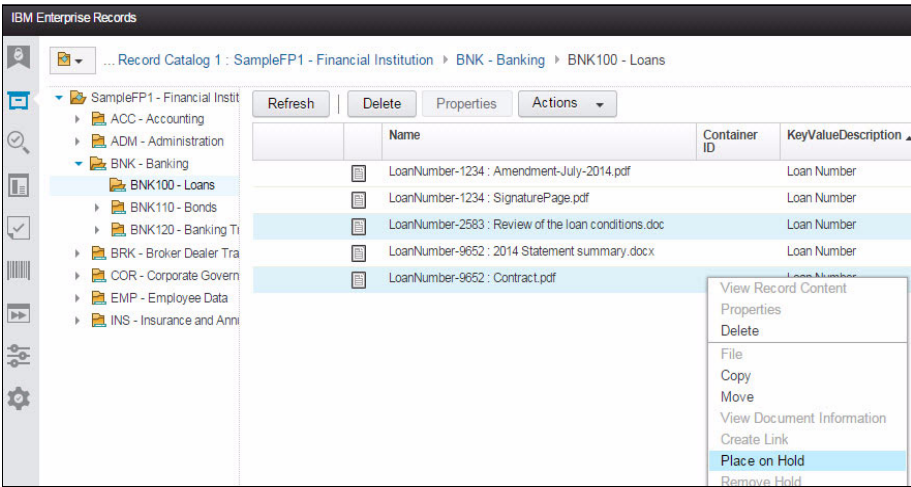


Figure 17-6 Selecting two entities to be placed on hold

3. Next, in the **Place Entities on Holds** page, from the list of available holds, select one or more holds that you want to place on the entities, and then click **Place on Hold**.

For this case study, we select **Investigation of Loan Practices** (Figure 17-7).

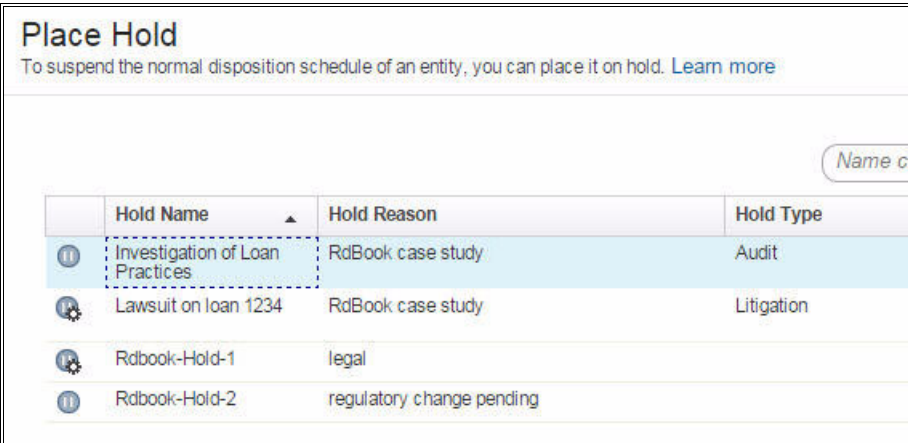


Figure 17-7 Select the holds to be placed on the selected records

4. Click **Hold**, and then, click **OK**.

Figure 17-8 shows the results of this case study. Notice that the two records now have the *on hold* symbols next to their entries.

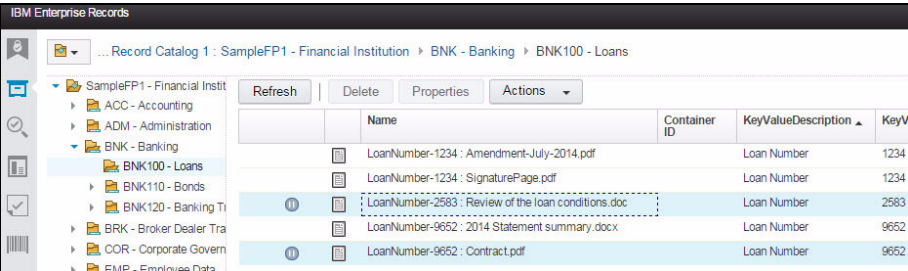


Figure 17-8 Records on hold

17.3.2 Removing a hold

If you place records on hold manually, you must remove them manually.

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 this case study, we navigate to **SampleFP1** → **BNK - Banking** → **BNK100 - Loans**.

2. Right-click the record to remove the hold, and select the **Remove Hold** action from the pop-up menu. See Figure 17-9.

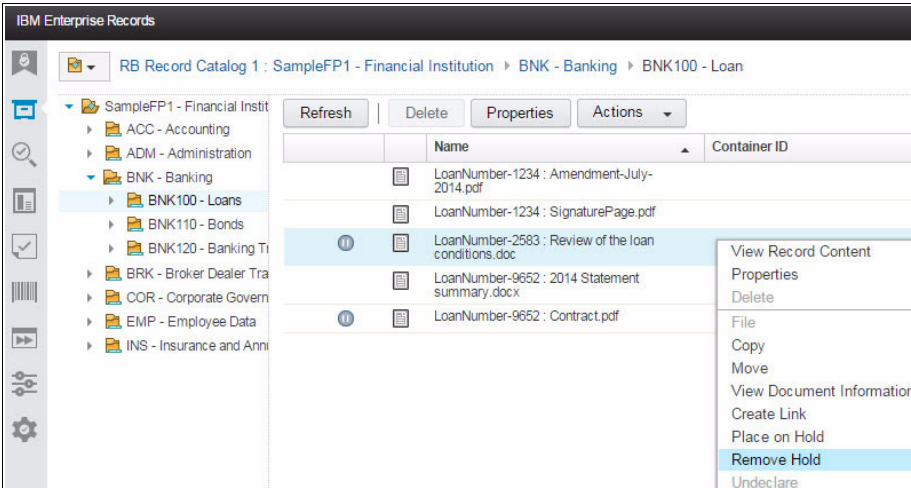


Figure 17-9 Select Remove Hold menu option

3. From the Remove Hold dialog window, select the hold that you want to remove. It is possible for an entity to have multiple holds on it for multiple litigation or various business reasons. In this window, you can remove one or multiple holds on this entity manually.

For this case study, we have only one hold. Select that hold and click **Remove Hold** (Figure 17-10).

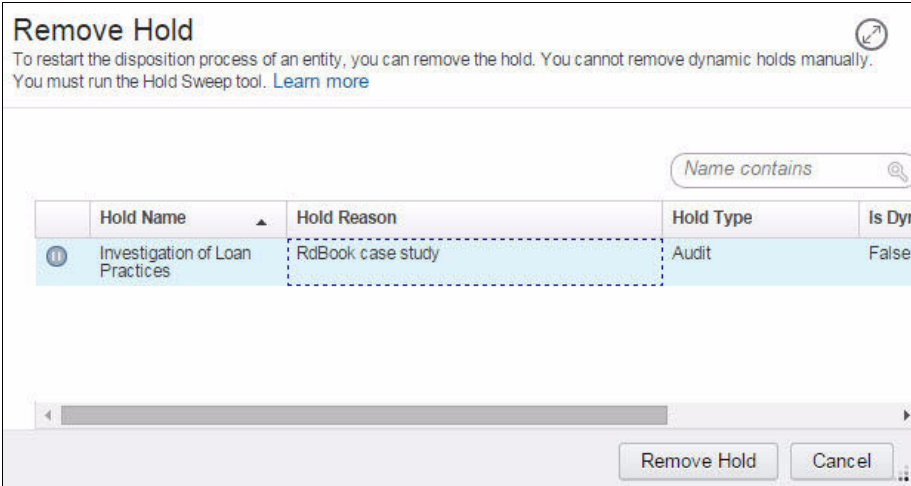


Figure 17-10 Selecting the hold to remove

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 **Open Configuration View** icon and **Holds** from the drop-down menu. Right-click the hold from which you want to remove the associated records and select the **Review Entities on Hold**, or select the hold and then click **Review Entities on Hold**, as shown in Figure 17-11 on page 372

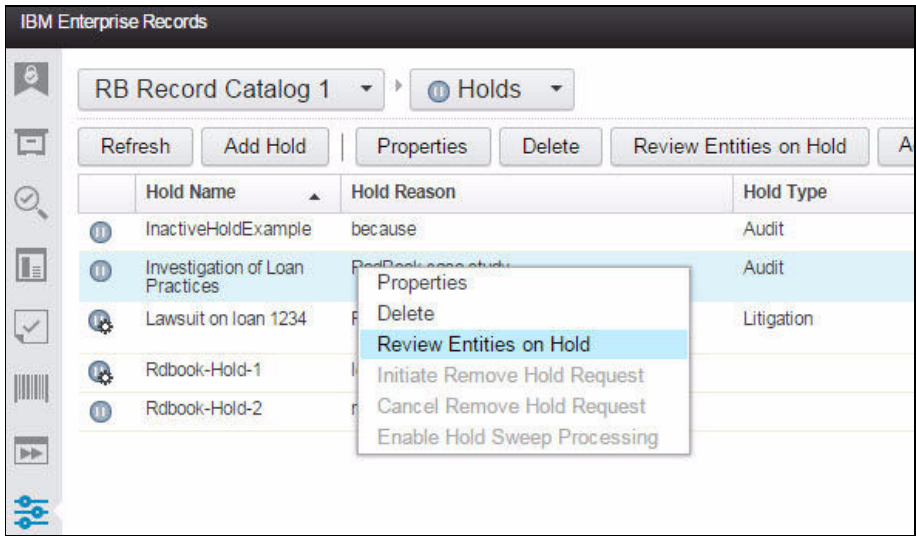


Figure 17-11 Launching the Review Entities on Hold action

2. In the **Review Entities on Hold** dialog, search for the records to remove. For this case study, just click **Search** because we only have one record associated with that hold, there is no need to provide any search parameter. Select the records to remove from the hold and click **Remove Hold**, or right-click the record and select the **Remove Hold** pop-up menu action, as shown in Figure 17-12.

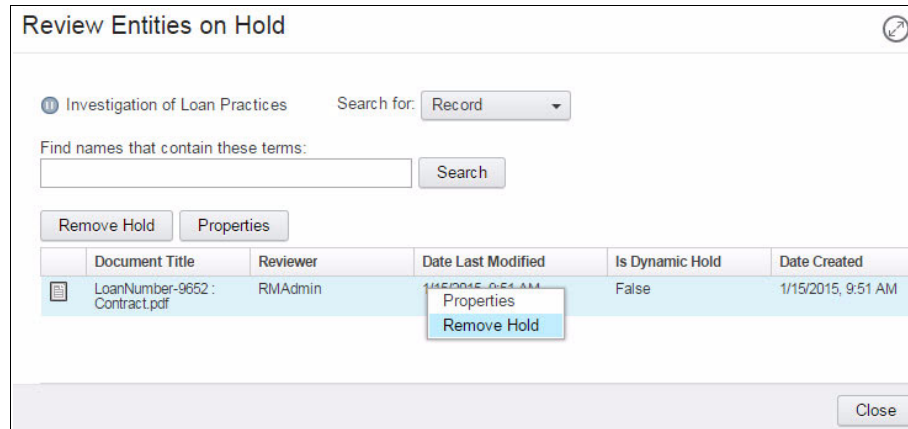


Figure 17-12 Review Entities on Hold

3. **Close** the dialog panel when all of the records are removed from that hold.

17.4 Dynamic holds and Hold Sweep

Hold Sweep is responsible for finding records that meet the conditions specified in conditional holds and then placing the hold on the records. In this case study, the Lawsuit on loan 1234 hold is dynamic and can be used with Hold Sweep.

A dynamic hold has the following lifecycle:

1. Active Hold Sweep: Pending

A dynamic hold was just created or reactivated (that occurs when a dynamic hold has gone through its lifecycle and is reactivated for a second cycle). Hold Sweep has never run for that hold. No record is put on hold yet.

2. Active Hold Sweep: Started

Hold sweep has been run at least once. Every time the Hold Sweep runs, it adds newly created records that match the condition on hold.

3. Requested Hold Removal

The hold is no longer needed and a request has been made to remove all of the holds for that dynamic hold from the records.

4. Removed Hold Sweep

Hold Sweep has run after a Requested Hold Removal. It is now inactive from a dynamic hold perspective, so it will not put any record on hold again until a request is made to reactivate it.

There are two ways to launch Hold Sweep:

- ▶ As a batch process
- ▶ As a task in IBM Content Navigator Task Manager

17.4.1 Launching Hold Sweep as a batch process

When run as a batch process, Hold Sweep must be launched using the operating system command line. It can also be scheduled by using the operating system scheduler to run when the system load is lower. Before you can run Hold Sweep as a batch process, you must configure it for the appropriate values.

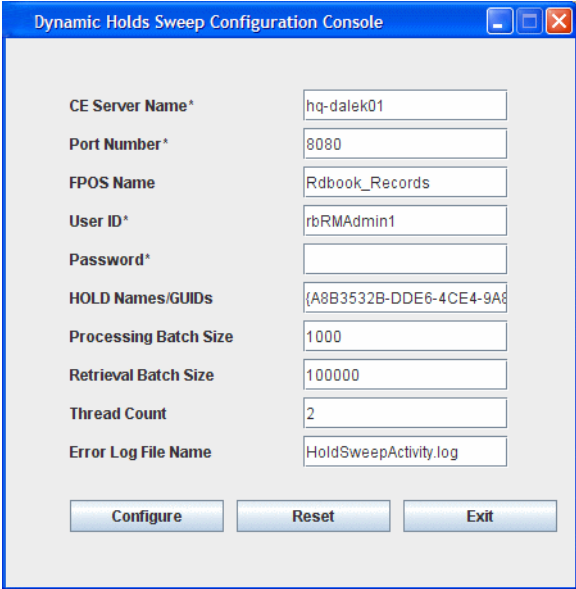
Configuring Hold Sweep

You can configure Hold Sweep to run against a specific file plan object store (FPOS) or all of the FPOSes in a Content Engine. To configure Hold Sweep, perform the following tasks:

1. From a command prompt, navigate to the RecordsManagerSweep folder (typically installed under <IBM Enterprise Records Installation Folder>\RecordsManagerSweep) and run the following command:

```
RecordsManagerSweep.bat -HoldSweep -configure
```

The Dynamic Holds Sweep Configuration Console dialog panel opens (Figure 17-13).



The image shows a Windows-style dialog box titled "Dynamic Holds Sweep Configuration Console". It contains several labeled text input fields arranged in a list. The fields and their values are: "CE Server Name*" with "hq-dalek01", "Port Number*" with "8080", "FPOS Name" with "Rdbook_Records", "User ID*" with "rbRMAAdmin1", "Password*" which is empty, "HOLD Names/GUIDs" with "{A8B3532B-DDE6-4CE4-9A8", "Processing Batch Size" with "1000", "Retrieval Batch Size" with "100000", "Thread Count" with "2", and "Error Log File Name" with "HoldSweepActivity.log". At the bottom of the dialog are three buttons: "Configure", "Reset", and "Exit".

Field	Value
CE Server Name*	hq-dalek01
Port Number*	8080
FPOS Name	Rdbook_Records
User ID*	rbRMAAdmin1
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 17-13 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 this case study, it is hq-da1eq01.

- Port Number: The Web Services Interface (WSI) port number that is used by your Content Engine server.

The default port number for Content Engine running under IBM WebSphere Application Server is 9080, WebLogic is 7001, and JBoss is 8080.

For this case study, we use 8080.

- FPOS NAME (optional field): Use the 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* of 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 rather than the name.

For this case study, it is RdBook_Records.

The GUID here is the 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 Records Administrator privileges.

For this case study, we use rbRMAAdmin1.

- Password: Password for the user ID.

- Hold Names/GUIDs: The name or GUID of up to five holds, separated by the | character (pipe, or vertical line).

The Hold Sweep process uses only the specified holds. If no holds are specified, the Hold Sweep processes *all of the active holds*.

For this 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 this 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 this 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: Enter the name and path of the error file to be created by the Hold Sweep process or use the default.

By default, a file called `HoldSweepActivity.log` is created in the `../FileNet/RecordsManagerSweep` folder.

For this case study, we use the default.

3. Click **Configure**. You will see a message indicating the successful configuration of Hold Sweep.

Running Hold Sweep

Assuming that you have created an active dynamic hold, you can automatically place or remove records on dynamic 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 run 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.

17.4.2 Launching Hold Sweep with Content Navigator Task Manager

Hold Sweep must be configured first to be usable under Content Navigator Task Manager.

Configuring Hold Sweep for Task manager

Task manager uses the Enterprise JavaBean (EJB) protocol to communicate with the sweep batch process. By default, Hold Sweep is configured for WSI.

1. To configure Hold Sweep for EJB, update the recordsmanagersweep.bat file (or recordsmanagersweep.sh for UNIX based system) where Hold Sweep is installed. Edit the following lines:

```
set CONNECTION_TYPE=WSI
set APP_SERVER=WebSphere

to

set CONNECTION_TYPE=EJB
set APP_SERVER=<your application server>
```

2. Go to the section that matches your application server and follow the instructions to adjust that section to your environment.
3. Using the IBM Enterprise Records application, set the location of the sweep batch process as shown in Figure 17-14 on page 377.

Select **Open Administration View**, expand the **Desktops** node in the tree view, and select your desktop. Select the **Disposition & Hold Sweep** tab and provide the information that includes the location of the sweep batch process.

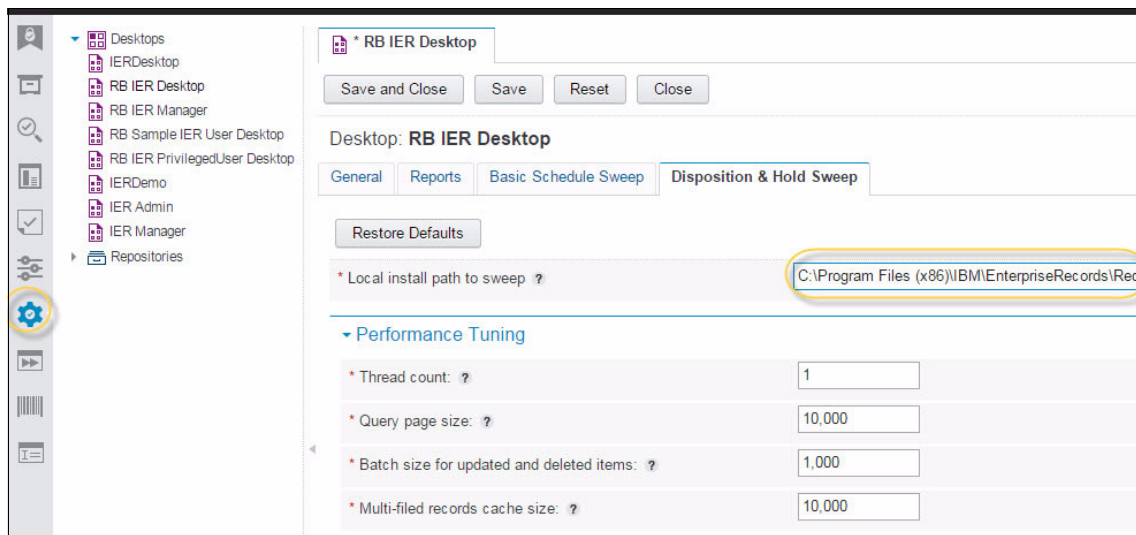


Figure 17-14 Set up the sweep process in the configuration view

Launching a hold task in Task Manager

To create a new Hold Sweep task, complete the following steps:

1. Select the **Open Task View** icon, click **Schedule** from the drop-down menu, and select the **Schedule Hold Sweep** action, as show in Figure 17-15.

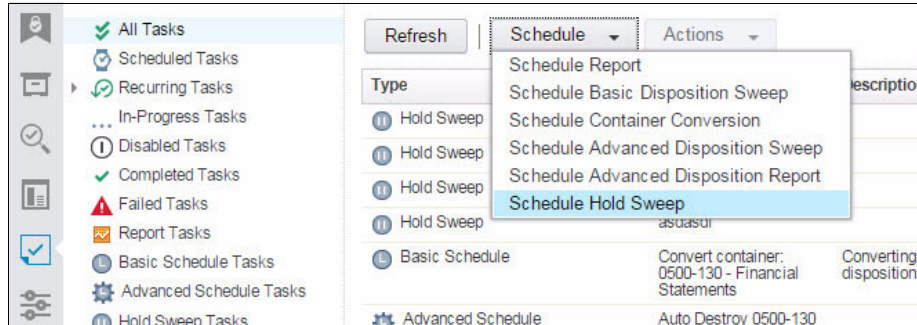


Figure 17-15 Launch the Schedule Hold Sweep task

2. In the task creation dialog panel, provide your repository name (the folder to where Hold Sweep will write the log) and a list of holds that you want this task to process. Then, click **Next**.
3. In the next window, provide the name for the task, specify whether you want to run it once or as recurring, and provide the user name and password to use for the sweep to access the record. The username to provide must have read/write access to all objects that it needs to process. Click **Schedule sweep** to launch the task.

17.4.3 Hold status changes

After Hold Sweep runs, the hold Sweep State for the hold changes to “Active holds sweep: started,” as shown in Figure 17-16.

The screenshot shows the 'Holds' table in the IBM Enterprise Records interface. The table has columns: 'Hold Name', 'Hold Reason', 'Hold Type', 'Active', and 'Sweep State'. There are five rows of data. The 'Sweep State' column for the third row, 'Lawsuit on loan 1234', is highlighted with a yellow circle and contains the text 'Active hold sweep: started'.

Hold Name	Hold Reason	Hold Type	Active	Sweep State
InactiveHoldExample	because	Audit	False	Manual hold
Investigation of Loan Practices	RdBook case study	Audit	True	Manual hold
Lawsuit on loan 1234	RdBook case study	Litigation	True	Active hold sweep: started
Rdbook-Hold-1	legal		True	Removed hold sweep
Rdbook-Hold-2	regulatory change pending		True	Manual hold

Figure 17-16 Hold Sweep State after the Hold Sweep has run at least once

This state means that Hold Sweep has run at least one time. Each time that the Hold Sweep runs for that hold, it obtains any new records that match the condition and puts them on hold also.

Note: If a record property changes so that it does not match the dynamic hold condition, Hold Sweep will not remove the hold from that record. Hold Sweep places only new records on hold.

Note: Changing the Active status of a hold (by updating the hold properties) to False results in Hold Sweep not taking any action on that hold (neither adding new records nor removing records on hold).

17.4.4 Verifying the records that are placed on hold

After you run the sweep (as a batch process or by using Content Navigator Task Manager), verify that the records are held as a result of running Hold Sweep:

1. Navigate to the entity from which you want to check the hold.
For this case study, we navigate to **SampleFP1** → **BNK - Banking** → **BNK100 - Loans**.
2. Verify that the hold icon is displayed next to the two records related to loan 1234, as shown in Figure 17-17.

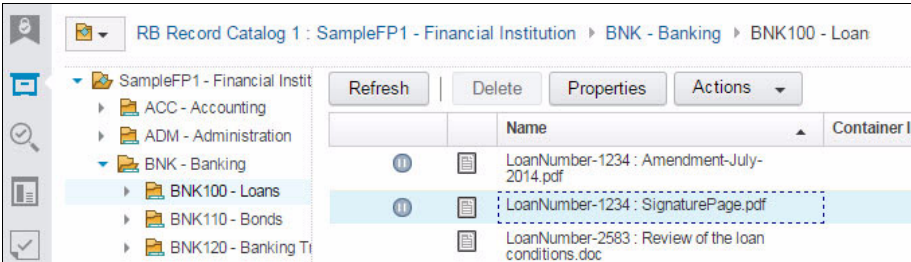


Figure 17-17 Show record placed on hold by the Hold Sweep

3. You can view which hold is placed on a record from the Hold tab of the record properties. Select a record and click **Properties** (or right-click the record and select the **Properties** action from the menu), as shown in Figure 17-18.

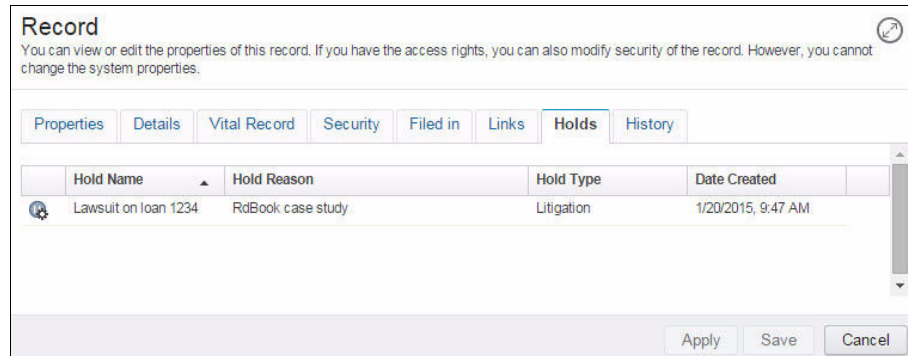


Figure 17-18 Review which hold is placed on a record

17.4.5 Removing dynamic holds using Hold Sweep

When the holds are no longer required, initiate a Remove Hold Request and then run Hold Sweep, using the same name of the hold.

Note: Using Hold Sweep is the only way to remove a dynamically applied hold from a record. However, if a dynamic hold (meaning a hold with a condition) was applied manually to a record, that hold can be removed manually. It will also be removed automatically by a sweep during the Initiate Remove Hold Request action described in this section.

To remove a hold from an entity, follow these steps:

1. Launch **IBM Enterprise Records Manager**.
2. Select the hold that you want to process. Select the **Open Configuration View** icon and the **Holds** drop-down menu. Right-click the hold from which you want to remove the associated records and select the **Initiate Remove Hold Request** option from the menu, as shown in Figure 17-19.

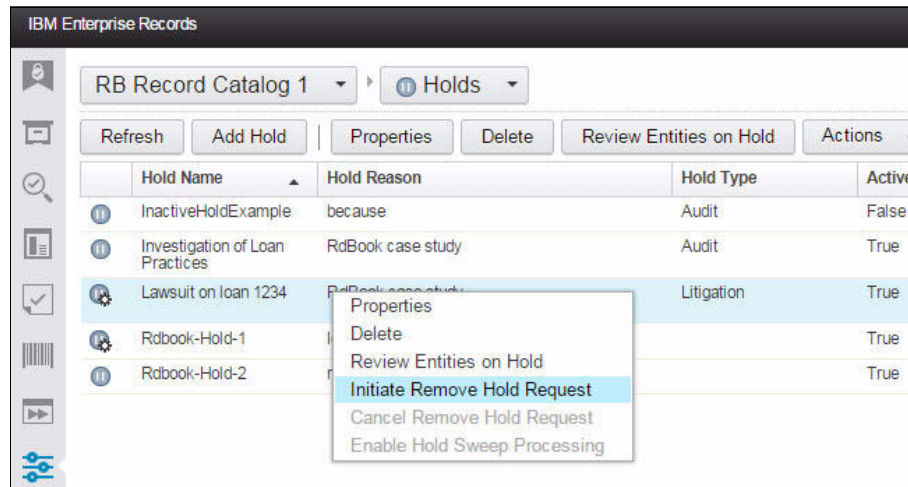


Figure 17-19 Initiate a Remove Hold Request

3. Notice that the hold status has changed to Requested Hold Removal, as shown in Figure 17-20.

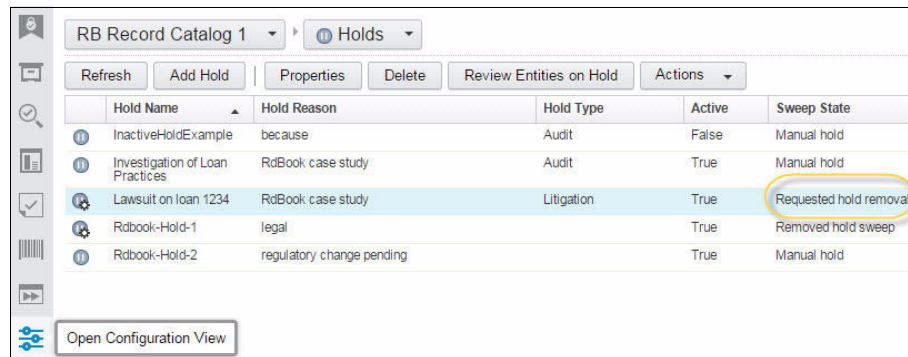


Figure 17-20 Hold status changed to Requested Hold Removal

4. Run Hold Sweep again (either as a batch process using the `RecordsManagerSweep.bat -HoldSweep` command or by using Content Navigator Task Manager) so that the system removes the hold from the corresponding entities.
5. Verify the results. Hold Sweep finds the records that are on hold due to the specific hold (for this case study, Lawsuit on loan 1234) and removes the holds from the records. Navigate to the category where the records are filed (for this case study, we navigate to **SampleFP1** → **BNK - Banking** → **BNK100 - Loans**) and verify that the hold icon has been removed. You might

need to click **Refresh** to refresh the list if you have not started a new browser session. See Figure 17-21.

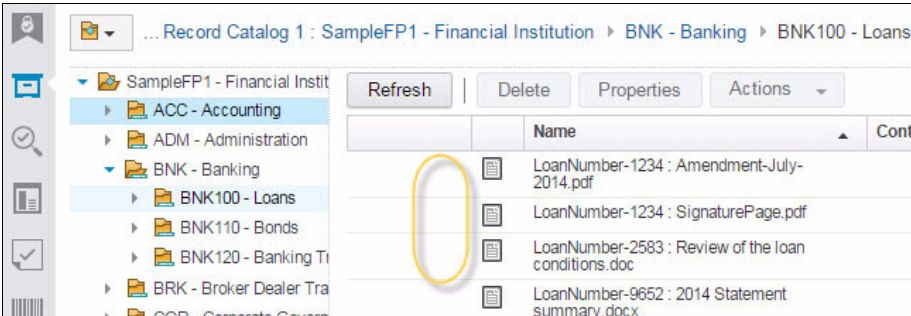


Figure 17-21 Holds have been removed from the records

6. The Sweep State of the hold has also been updated to **Removed Hold Sweep**, as shown in Figure 17-22. Any subsequent Hold Sweep run ignores that dynamic hold. No new record will be put on hold based on this hold condition.

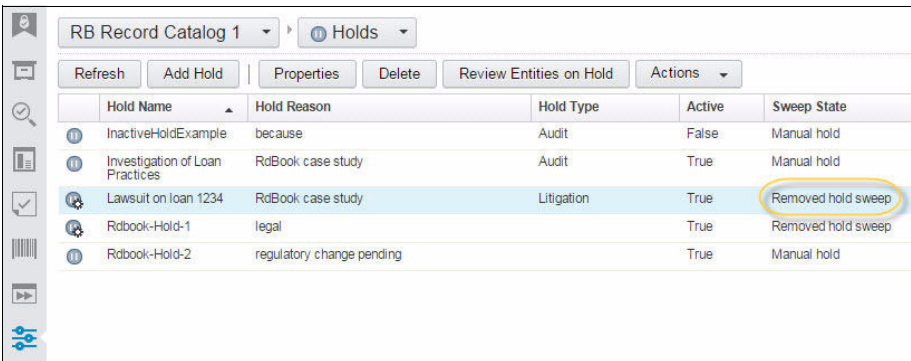


Figure 17-22 Sweep State after the hold have been removed

However, this dynamic hold can be reactivated by using the **Enable Sweep Processing** pop-up menu action and right-clicking the hold, as shown in Figure 17-23. That will return the hold state back to **Active Hold Sweep: Pending**.

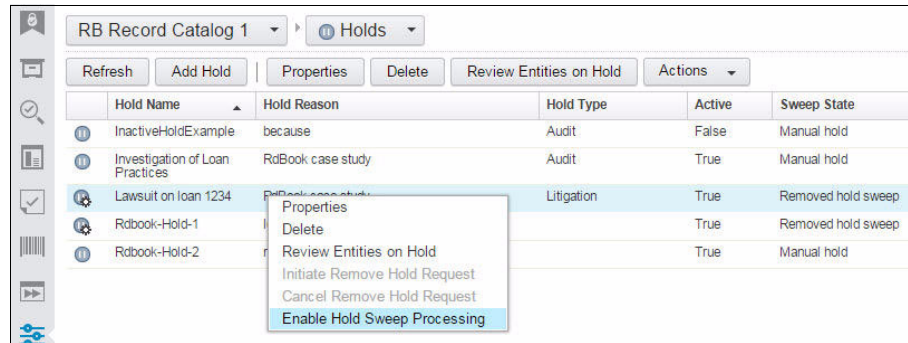


Figure 17-23 Re-enable a dynamic hold that was already removed



IBM Java API for Records Manager case study

This chapter provides sample code to illustrate the IBM Java API for Records Manager (JARM) programming capabilities in the context of records declaration.

Through use cases, this chapter describes implementing a batch program that declares newly added documents as records, based on certain custom properties, and implementing a custom event handler that maintains the location of those records in the file plan when the source document coding changes.

Note: The sample code provided here is merely to illustrate JARM. It is not optimized for performance.

This chapter covers the following topics:

- ▶ Description of the use cases
- ▶ Record populating batch sample code
- ▶ Event handler for record maintenance sample code

18.1 Description of the use cases

Two use cases are illustrated in this chapter:

► Use Case 1

This use case illustrates a schedule aggregation *record* solution. All of the records that belong to a specified record series category are filed directly under this category.

The category is set up with either a simple disposition schedule or an advanced disposition schedule aggregation record. Each record is disposed of independently.

► Use Case 2

This use case illustrates a schedule aggregation *folder* solution. The records are grouped under a subfolder of the record series category. Each subfolder represents a specific instance of the entity that the record belongs to. For example, each folder can represent an employee ID. The sample code creates the subfolder when needed if it does not already exist.

The category is set up with an advanced disposition schedule aggregation folder so that all of the records that belong to the same subject (for example, the same employee) are disposed of simultaneously.

Figure 18-1 shows record categories of the use cases.

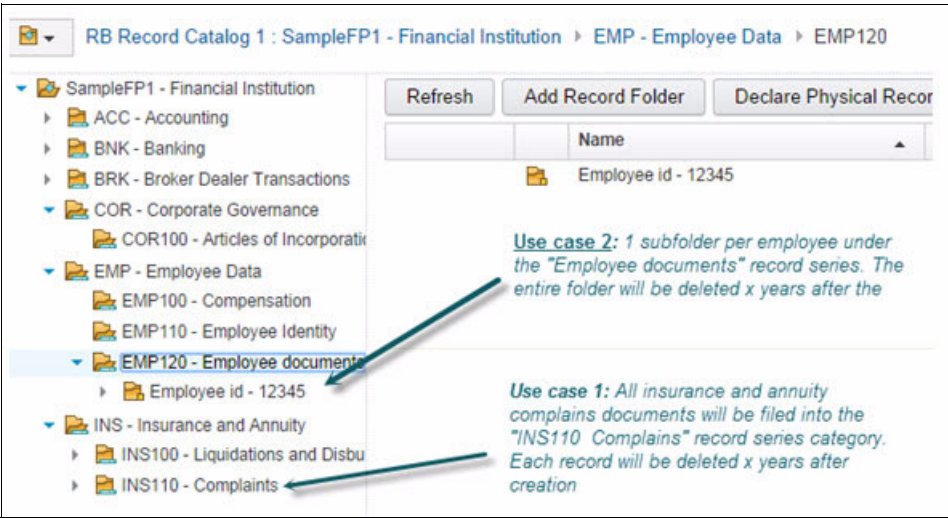


Figure 18-1 Use Cases 1 and Use Case 2

18.1.1 Content Engine class and properties setup

This sample solution requires the creation of custom classes and properties in the Content Process Engine, on both the record object store (ROS) and the file plan object store (FPOS). This section describes what classes and properties are needed.

- Create a Document subclass for the new documents.

We create one Document subclass to be used for both Use Case 1 and Use Case 2.

For the record object store, create a new subclass of the Document class, called `ILGDocument`, and add the following four properties:

- `RecordSeriesCode` (String): Requires for both use cases. This property specifies in which record series container the record should be filed.
- `RetentionTriggerDate` (Date): Requires only for Use Case 1. This date property specifies when to dispose of the record. It is not used in Use Case 2 because that information is defined at the folder level.
- `KeyValueDescription` (String): Requires only for Use Case 2. This property provides a description of the `KeyValue` for a given document, for example, Employee ID or Contract Number.
- `KeyValue` (String): Required only for Use Case 2. This property specifies the value of the entity that the document is for. For example, EMP1236 if the document is related to the employee whose ID is EMP1236 or CONT1417 if the document is related to the contract CONT1417.

`KeyValueDescription` and `KeyValue` are used in test case 2 to build the folder name that will gather all documents related to the same subject.

- Create a Record subclass that the sample code will use when declaring a document as record.

This subclass contains the `RetentionTriggerDate` that is propagated from the document to the record.

In this example, this subclass is required for Use Case 1 only:

On the file plan object store (FPOS), create a new subclass of the Electronic Record class, `ILGRecord`, and add the following property:

- `RetentionTriggerDate` (Date): This date property specifies when to dispose of the record. It is defined on the source document and is auto-synchronized to the record by using the sample synchronize event.

Note: For the auto-synchronization event handler to work, make sure to use the same symbolic name for the property template in the ROS and the FPOS.

- Create a Record Folder subclass that the population sample code will use when creating new folder when needed to file the records.

This is required only for Use Case 2 because Use Case 1 does not create subfolders.

For the file plan object store, create a new subclass of the Electronic Record Folder class, `ILGFolder`, and add the following properties:

- `KeyValueDescription` (String): This property provides a description about the `KeyValue` for a given document, for example, Employee ID or Contract Number.
- `KeyValue` (String): This property specifies the value of the entity that the documents are for.
- `RetentionTriggerDate` (Date): This date property specifies when to dispose of the folder.

- Define `AggregationType` property in the Record Category class.

The Record Category class must have the `AggregationType` property defined for the sample code to identify which use case the Record Category is for.

This property must be defined for both Use Case 1 and Use Case 2.

For the file plan object store, update the `RecordCategory` class and add the following property:

- `AggregationType` (String): This property specifies whether this record series category behaves as Use Case 1 or Use Case 2. It is possible to associate the `AggregationType` with a choice list to ensure that only one of the two values, `RecordFolder` or `Record`, can be used.

Note: This sample code uses this property value on the category to determine the behavior about how to file the record. Another approach is to traverse the disposition schedule associated with this category to determine its aggregation level. For a simple disposition schedule, the aggregation is always `Record`.

18.1.2 Use Case 1 walk-through

Use Case 1 illustrates a schedule aggregation record solution. In this type of schedule, the records are deleted individually on a per-record basis. They are all filed under the same record series category. The schedule associated with the record series category can be either a simple schedule or an advance schedule aggregation record.

When creating the record series category, provide at least a name, the record series identifier code in the Record Category Identifier property, and the Aggregation type, Record, as illustrated in Figure 18-2. The Record aggregation identifies the category as behaving as Use Case 1.

The screenshot shows a web form for creating a 'Record Category'. The 'Class' dropdown is set to 'Record Category'. The 'Record Category Name' field contains 'INS110 - Complaints'. The 'Record Category Identifier' field contains 'INS110'. The 'Description' field is empty. The 'Date Opened' field shows '9/22/2014, 11:28 AM' with a calendar icon. The 'Reviewer' field shows 'RMAdmin'. The 'AggregationType' field is set to 'Record'. Annotations include a green arrow pointing from 'Record Series code' to the 'Record Category Identifier' field, and another green arrow pointing from 'Aggregation type : Record' to the 'AggregationType' field.

* Class:	Record Category
* Record Category Name: ?	INS110 - Complaints
* Record Category Identifier: ?	INS110
Description: ?	
Date Opened: ?	9/22/2014, 11:28 AM
* Reviewer: ?	RMAdmin
AggregationType: ?	Record

Figure 18-2 : Creating a record series category for Use Case 1

When adding a new document in Content Navigator, select the **ILGDocument** class, and provide a value for the RecordSeriesCode and the RetentionTriggerDate (optional), as illustrated in Figure 18-3 on page 390.

The record series code on the document must match the Record Category Identifier set on the record series category. This is the property that the population sample code uses to determine where to file the record.

The RetentionTriggerDate is optional because it might not be known when the document is created.

Properties *Record series code*

*Record Category Name: ? EMP120 - Employee documents

*Record Category Identifier: ? EMP120

Description: ? Monitoring and reporting on all employee related activity and activity of associated persons such as: Employee Files, Fingerprinting, Compensation, Benefits, Registration and Licensing

Date Opened: ? 9/25/2014, 8:37 AM

*Reviewer: ? P8Admin

AggregationType: ? RecordFolder *Aggregation type: RecordFolder*

Figure 18-4 Creating a record series category for Use Case 2

When adding a new document in IBM Content Navigator, select the **ILGDocument** class, and provide a value for the RecordSeriesCode, KeyValue, and KeyValueDescription properties as illustrated in Figure 18-5.

Properties

* Class: ILG Document

Document Title: ?

DocumentDate: ? M/d/yyyy, h:mm a

DocumentType: ?

KeyValue: ? 12345 *Employee id, which will create a subfolder if needed*

KeyValueDescription: ? Employee id *Record series ID where to file the record*

RecordSeriesCode: ? EMP120

RetentionTriggerDate: ? M/d/yyyy, h:mm a

OrganizationUnit: ?

LocalScheduleID: ?

Figure 18-5 Adding a document

The fields are used as follows:

- The record series code on the document must match the “Record Category Identifier” set on the record series category “Record Category Identifier” property to where to file the record. This is the property that the population sample code uses to determine where to file the record.

- ▶ The KeyValue property provides the value for the subject that will be used to file the record into the correct subfolder (or create this subfolder dynamically if needed).
- ▶ The KeyValueDescription specifies what the KeyValue is.
- ▶ The RetentionTriggerDate is not needed here, because that is information that will be managed on the folder later when the external event occurs.

After the population sample code runs and declares the new documents as records, a new folder is created, if require, for all new record subject, as illustrated in Figure 18-6.

General

* Save in: EMP120 - Employee documents

Properties

* Class: ILGFolder

*Record Folder Name: ? Employee id: 12345

*Folder Unique Identifier: ? Employee id: 12345

Description: ? Monitoring and reporting on all employee related activity and activity o Files; Fingerprinting; Compensation, Benefits, Registration and Licens

Date Opened: ? 9/25/2014, 8:40 AM

*Reviewer: ? P8Admin

KeyValue: ? 12345

KeyValueDescription: ? Employee id

RetentionTriggerDate: ? M/d/yyyy, h:mm a

Employee id: All documents related to this employee will be filed into this folder

ID of the employee

Figure 18-6 Folder created by the population sample code

All records belonging to the same KeyValue will be filed under that folder.

18.2 Record populating batch sample code

The sample code in Example 18-1 on page 393 illustrates the use of the Java API for Content Engine (JACE) and the Java API for Records Manager (JARM) to run a batch process that will declare newly created documents as records, as explained in 18.1, “Description of the use cases” on page 386.

See 12.3, “Records Manager API” on page 281 for a better understanding of the JARM API.

Example 18-1 Record population batch sample code

```
package com.ibm;

import java.util.*;
import com.filenet.api.collection.*;
import com.filenet.api.core.*;
import com.filenet.api.query.*;
import com.filenet.api.util.*;
import com.ibm.jarm.api.collection.*;
import com.ibm.jarm.api.constants.*;
import com.ibm.jarm.api.core.*;
import com.ibm.jarm.api.property.*;
import com.ibm.jarm.api.query.*;
import com.ibm.jarm.api.security.*;
import com.ibm.jarm.api.util.*;

public class RecordDeclarationSweep {

    private static final String JAAS_STANZA = RMUserContext.P8_STANZA_WSI;

    // IER JARM objects
    private static DomainConnection jarmDomainConnection;
    private static RMDomain jarmDomain;
    private static ContentRepository jarmROS;
    private static FilePlanRepository jarmFPOS;

    // CE JACE Objects
    public static Connection jaceDomainConnection;
    public static Domain jaceDomain;
    private static ObjectStore jaceOS;

    private static String userName;
    private static String password;
    private static String ceServerName;
    private static String ceWSIPortNumber;
    private static String fposName;
    private static String rosName;
    private static String lastSweepDate;

    public static void main( String [] args ) throws Exception
    {

        // Read the system variable parameters from an external properties file
        userName      = "<rmAdminUser>";
        password       = "<rmAdminPassword>";
        ceServerName   = "<CEServerName>";
        ceWSIPortNumber = "<CEPortNumber>";
        fposName        = "<FPosName>";
        rosName         = "<RosName>";
        lastSweepDate  = "<LastTimeTheProgramWasRun (YYYY-MM-DD)";
```

```

        // Compute names
        String ceServerURL = "http://" + ceServerName + ":" + ceWSIPortNumber +
"/wsi/FNCEWS40MTOM";

        // Create a JARM connection to the CE
        jarmDomainConnection =
RMFactory.DomainConnection.createInstance(DomainType.P8_CE, ceServerURL, null);

        // Set the IER subject
        com.ibm.jarm.api.util.RMUserContext ierUC =
com.ibm.jarm.api.util.RMUserContext.get();
        javax.security.auth.Subject subject =
com.ibm.jarm.api.util.RMUserContext.createSubject(jarmDomainConnection, userName,
password, JAAS_STANZA);
        ierUC.setSubject(subject);

        // Get the IER JARM domain
        jarmDomain = RMFactory.RMDomain.fetchInstance(jarmDomainConnection, null, null);

        // Connect to the IER object stores
        jarmROS =
com.ibm.jarm.api.core.RMFactory.ContentRepository.fetchInstance(jarmDomain, rosName,
null);
        jarmFPOS = RMFactory.FilePlanRepository.fetchInstance(jarmDomain, fposName,
null);

        // Create a JACE connection to the CE
        jaceDomainConnection = Factory.Connection.getConnection(ceServerURL);

        // Set the CE subject
        UserContext ceUC = UserContext.get();
        javax.security.auth.Subject subject =
UserContext.createSubject(jaceDomainConnection, userName, password, JAAS_STANZA);
        ceUC.pushSubject(subject);

        // Get the CE JACE domain
        jaceDomain = Factory.Domain.fetchInstance(jaceDomainConnection, null, null);

        // Connect to the ROS object stores
        jaceOS = Factory.ObjectStore.fetchInstance(jaceDomain, rosName, null);

        // Get all the newly created documents not yet declared as record and declare
them
        RepositoryRowSet rowSet = returnNewDocuments();

        // Loop on the document list and declare each document as record
        Iterator iter = rowSet.iterator();
        while (iter.hasNext())
        {
            try
            {
                RepositoryRow row = (RepositoryRow) iter.next();

```

```

        // Get necessary documents properties
        String recordSeriesCode =
row.getProperties().get("RecordSeriesCode").getStringValue();
        String keyValue =
row.getProperties().get("KeyValue").getStringValue();
        String KeyValueDescription =
row.getProperties().get("KeyValueDescription").getStringValue();

        // Do not declare a document that has no record series code
        if (recordSeriesCode != null )
        {
            // Get the record series category which RecordCategoryIdentifier is
equal to the recordSeriesCode value
            RecordCategory recordSerieCategory =
returnRecordSeriesContainer(recordSeriesCode);

            // Retrieve the aggregation type of the record series category
            String aggregationType =
recordSerieCategory.getProperties().getStringValue("AggregationType");

            // Check the custom property "AggregatinType" set on the record series
container that defines how the record will be declared
            // - AggregationType = "Record" => The record is declared directly
under the record series category
            // - AggregationType = "RecordFolder" => The record is declared under
a sub-folder of the record series container
            if (aggregationType.equalsIgnoreCase("RecordFolder"))
            {
                if (keyValue != null && KeyValueDescription != null)
                {
                    // Get the record sub-folder where to declare the record to. If the
record folder doesn't exists, create it on the fly
                    RecordFolder recordSeriesSubFolder =
returnRecordSeriesSubFolder(recordSerieCategory, keyValue, KeyValueDescription);

                    // Declare the document as record in the record sub-folder
                    declareDocumentAsRecord(row, recordSeriesSubFolder,
aggregationType);
                }
            }
            else
            {
                // Declare the document as record directly into the record series
                declareDocumentAsRecord(row, recordSerieCategory, aggregationType);
            }
        }
    }
    catch (Exception e)
    {
        // Something didn't work well
        System.out.println(e.getMessage());
    }
}

```

```

    }

    /*
     * Declare a document as record
     */
    private static void declareDocumentAsRecord(RepositoryRow row, RecordContainer
container, String aggregationType) throws Exception
    {
        String docId =
row.getProperties().get("Id").getIdValue().toString();
        String docTitle =
row.getProperties().get("DocumentTitle").getStringValue();
        Date retentionTriggerDate =
row.getProperties().get("RetentionTriggerDate").getDateTimeValue();
        Date documentDate =
row.getProperties().get("DocumentDate").getDateTimeValue();

        // Define properties for the new record.
        RMProperties jarmProps = RMFactory.RMProperties.createInstance(DomainType.P8_CE);
        jarmProps.putStringValue(RMPropertyName.DocumentTitle, docTitle);
        jarmProps.putDateTimeValue("DocumentDate", documentDate);
        if (aggregationType.equalsIgnoreCase("Record"))
            jarmProps.putDateTimeValue("RetentionTriggerDate", retentionTriggerDate);
        // No additional permission specified.
        List<RMPermission> jarmPerms = null;
        // No additional record filings needed.
        List<RecordContainer> additionalContainers = null;
        // Collection of content to declare for.
        List<String> targetDocIDs = new ArrayList<String>(1);
        targetDocIDs.add(docId);

        // Ready to perform new electronic record declaration.
        container.declare("ILGRecord", jarmProps, jarmPerms, additionalContainers,
jarmROS, targetDocIDs);
    }

    /*
     * Return the list of newly create ILG documents that have not been declared as
record already
     */
    private static RepositoryRowSet returnNewDocuments()
    {
        // Build the SQL statement
        String mySQLString = "SELECT Id, DocumentTitle, DocumentDate, DocumentType,
KeyValue, KeyValueDescription, RecordSeriesCode, RetentionTriggerDate, OrganizationUnit,
LocalScheduleID FROM ILGDocument WHERE DateCreated > " + lastSweepDate + " and
RecordInformation is null";
        SearchSQL sqlObject = new SearchSQL();
        sqlObject.setQueryString(mySQLString);
        SearchScope searchScope = new SearchScope(jaceOS);
        return searchScope.fetchRows(sqlObject, null, null, new Boolean(true));
    }
}

```



```

        /* Illustrate a none paging search of a category
        * Search for a category with Record Category Identifier equals to the recordSerieId
parameter
        */
        private static RecordCategory returnRecordSeriesContainer(String recordSeriesCode)
throws Exception
        {

            // ---- Perform a search that returns ResultRows ----
            RMSearch jarmSearch = new RMSearch(jarmFPOS);
            // P8 SQL statement that includes a JOIN clause.
            // Note use of column alias 'DateRecFiled'.
            String sqlStmt = "SELECT rc.Id, rc.RecordCategoryName,
rc.RecordCategoryIdentifier FROM RecordCategory rc WHERE rc.RecordCategoryIdentifier =
'" + recordSeriesCode + "'";
            // No object values returned.
            RMPropertyFilter filter = null;
            // Not going to page.
            Integer pageSize = null;
            // Perform the search operation...
            Boolean continuable = Boolean.FALSE; // non-paging
            PageableSet<ResultRow> rowResultSet =
jarmSearch.fetchRows(sqlStmt, pageSize, filter, continuable);
            // Use normal iterator to access results in a non-paged manner.
            Iterator<ResultRow> iter = rowResultSet.iterator();
            int rowNumber = 0;
            String rcId = null;
            while ( iter.hasNext() )
            {
                ResultRow row = iter.next();
                RMProperties rowProps = row.getProperties();
                rcId = rowProps.getStringValue("ID");
                rowNumber ++;
            }

            if (rowNumber != 1)
                throw new Exception("We are expecting to find one and only one category with
identifier = '" + recordSeriesCode + "', but we found " + rowNumber);
            else
                return RMFactory.RecordCategory.fetchInstance(jarmFPOS, rcId, null);
        }

        /* Illustrate the getContent of a container
        * Search for a category with Record Category Identifier equals to the recordSerieId
parameter
        */
        private static RecordFolder returnRecordSeriesSubFolder(RecordCategory
recordSerieCategory, String keyVal, String keyValDescription) throws Exception
        {
            RecordFolder subRecordFolder = null;

            // Build a filter to get the list of custom properties

```

```

RMPPropertyFilter filter = new RMPPropertyFilter();
Integer maxRecursion = Integer.valueOf(1); // Recursion depth to go down.
Long maxContentSize = null; // No content involved - not applicable.
Boolean levelDependents = null; // Use the repository default setting.
Integer pageSize = null; // Use the repository default setting.
// List of property symbolic names to include.
String symbolicNames = "FolderName KeyValue KeyValueDescription";
// Desired result set page size.
filter.addIncludeProperty(maxRecursion, maxContentSize, levelDependents,
symbolicNames, pageSize);

// Get all folders created under the record serie category
// For demonstration purposes, use the paging mechanism of this PageableSet.
PageableSet<RecordFolder> rcResultSet =
recordSerieCategory.fetchRecordFolders(filter, pageSize);
RMPageIterator<RecordFolder> rcPI = rcResultSet.pageIterator();

// Loop on all the pages
while ( rcPI.nextPage() )
{
    // For each page, loop on the folders
    List<RecordFolder> rcPage = rcPI.getCurrentPage();
    for (RecordFolder recFolder : rcPage)
    {
        if (keyValue.equals(recFolder.getProperties().getStringValue("KeyValue")))
        {
            // Exit this loop if we found the folder
            subRecordFolder = recFolder;
            break;
        }
    }

    // Exit the pagination loop if we have already found the folder
    if (subRecordFolder != null)
        break;
}

// If the specified grant folder has not been found under the record serie
category, create it on the fly
if (subRecordFolder == null)
{
    RMPProperties jarmProps =
RMFactory.RMPProperties.createInstance(DomainType.P8_CE);
    jarmProps.putStringValue(RMPPropertyName.RecordFolderName, keyValueDescription
+ " - " + keyValue);
    jarmProps.putStringValue(RMPPropertyName.RecordFolderIdentifier, keyValue);
    jarmProps.putStringValue("keyValue", keyValue);
    jarmProps.putStringValue("KeyValueDescription", keyValueDescription);

    // Inherit the schedule from the parent container

    jarmProps.putObjectValue(com.ibm.jarm.api.constants.RMPPropertyName.DispositionSchedule,
recordSerieCategory.getAssignedSchedule());
}

```

```

jarmProps.putGuidValue(com.ibm.jarm.api.constants.RMPropertyName.DispositionScheduleInheritedFrom, recordSerieCategory.getObjectIdentity());

        subRecordFolder = recordSerieCategory.addRecordFolder("ILGFolder", jarmProps,
null);
    }

    // Return the grant folder where to file the record to
    return subRecordFolder;
}
}

```

18.3 Event handler for record maintenance sample code

When the document properties are updated, and the update occurs on one of the properties that affects the record file in location within the file plan, it is important to move the record to the new location.

This is implemented through a Content Engine (CE) document update event. We demonstrate how to create such an event handler, subscribe it to the Content Engine update event, and, if necessary, move the record, based on the new document property values.

The event code follows these rules:

- ▶ Get the record object from the updated document object passed by the CE.
- ▶ Get the record series category and potentially parent subfolders where the record is currently filed.
- ▶ Verify that the current location is still valid regarding the document updated properties. If the location has been changed (KeyValue or RecordSeriesCode properties are updated on the document), the sample code relocates the record to the new container, following the same rules.

The code presented here is an extension of the ready-for-use `AutoSyncProperties.java` event handler that is delivered with IBM Enterprise Records under this directory path:

`Events\src\com\filenet\rm\ceintegration\eventhandler`

This event handler automatically synchronizes changed properties with same symbolic name value from the document to the record when they are updated on the document. After the synchronization is complete, we add extra code to verify whether the record needs to be moved.

The synchronization code uses the native CE JACE API only, so it does not need any extra JAR files (the JACE JAR files are automatically added to the class path of the event handler by CE). However, the extra code implemented here is based on the JARM API. Therefore, we need to add those extra JARM JAR files to the event handler.

18.3.1 Update the existing AutoSyncProperties.java file

This section describes how to prepare a development environment and how to update the AutoSyncProperties.java code that is included with the software to add the new JARM-based maintenance sample code.

Set up the development environment

Follow these required steps to prepare your development environment to update the event handler:

1. Get the events folder from the Enterprise Records installation CD locally.
2. Create a new Eclipse project.
3. In this project, create the following folder package hierarchy **src** → **com** → **filenet** → **rm** → **ceintegration** → **eventhandler**.
4. Add the AutoSyncProperties.java file delivered under the Events folder from the installation CD to the EventHandler folder (see Figure 18-7 on page 401).
5. Add the RMAutoSynchronizePropertiesImport.xml file at the root or the project level.
6. Add a lib folder, and add the following files from your Enterprise Records environment:
 - Jarm.jar
 - JarmResources.jar
 - ierLogTrace.jar

You can find these files on the server where the IBM Enterprise Records application is installed:

<IBM Enterprise Records installation folder>\API\JARM

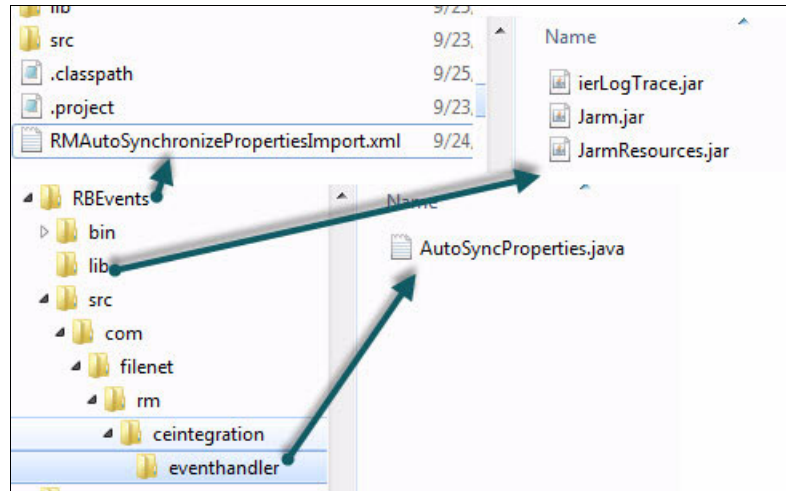


Figure 18-7 Event handler source hierarchy

Update the AutoSyncProperties.java source code

Update the AutoSyncProperties.java source code to add the new JARM-based maintenance sample code.

Complete the following steps to update the source code:

1. Update the doPropertiesSync() method to call the new maintainRecordLocation() method after the second synchronizeRecordDocumentProperties(), as shown in Example 18-2.

Example 18-2 Add the call to the maintainRecordLocation method

```
private void doPropertiesSync(Id docId, ObjectStore objStore, StringList
modifiedProps) throws Exception
{
    ...
    if (engObj instanceof Document)
    {
        Document destdoc = (Document) engObj;
        synchronizeRecordDocumentProperties(srcdoc, destdoc, modifiedProps);
        maintainRecordLocation(srcdoc, destdoc);
    }
    ...
}
```

2. Add the new maintainRecordLocation() method as shown in Example 18-3.

Example 18-3 Method to be added to the AutoSyncProperties.java file

```
/*
 * Method that check if a record needs to be moved to a different location
 * in the case that the source document properties have been changed
 */
private void maintainRecordLocation(Document jaceSourceDocument, Document jaceRecord)
throws Exception
{
    String recordId = jaceRecord.getId().toString();

    // Convert the JACE Object store where the record resides into JARM FPOS
    jarmFPOS = (FilePlanRepository)
com.ibm.jarm.api.util.P8CE_Convert.fromP8CE(jaceRecord.getObjectStore());

    // "Convert" the JACE record Document object into a JARM record Record object
    Record jarmRecord = RMFactory.Record.fetchInstance(jarmFPOS, recordId, null);

    // Verify if the record has all the properties required to manage its location
    String recordRecordSeriesCode = null;
    String recordKeyValue = null;
    String recordKeyValueDescription = null;
    try
    {
        recordRecordSeriesCode =
jaceSourceDocument.getProperties().get("RecordSeriesCode").getStringValue();
        recordKeyValue =
jaceSourceDocument.getProperties().get("KeyValue").getStringValue();
        recordKeyValueDescription =
jaceSourceDocument.getProperties().get("KeyValueDescription").getStringValue();
    }
    catch (Exception e)
    {
        // this record is missing a property and therefore can't be manage by this
algorithm
        return;
    }

    // Verify that the record's Record Series code is defined properly. If not, just
exit from here
    if (recordRecordSeriesCode == null) return;

    // Get the record parent containers hierarchy (folder if any, and record series
parent category)
    RecordCategory parentRecordSeries = null;
    RecordFolder parentFolder = null;
    RecordVolume parentVolume = null;

    // Make sure that the record is only filed in 1 container
    List<Container> parentContainers = jarmRecord.getContainedBy();
    if (parentContainers.size() == 1)
    {
```

```

RecordContainer parentContainer = (RecordContainer) parentContainers.get(0);

// if the record is filed directly into the record series category, then set
if (parentContainer instanceof RecordCategory)
{
    parentRecordSeries = (RecordCategory) parentContainer;
}
else
{
    // if the record is filed into a subfolder of the record series category, then
    set the entire hierarchy of parent containers
    parentVolume = (RecordVolume) parentContainer;
    parentFolder = (RecordFolder) parentVolume.getParent();
    parentRecordSeries = (RecordCategory) parentFolder.getParent();
}
}
else
{
    // If the record is multi-filed, then we can't manage it with this algorithm.
    Exit from here
    return;
}

// Get recordSeriesCode, keyValue and KeyValueDescription from the containers where
the record is currently filed into
// We will be able to compare it with the current record value to see if we need to
relocate the record.
String containerRecordSeriesCode =
parentRecordSeries.getProperties().get("RecordCategoryIdentifier").getStringValue();
String containerKeyValue = null;
String containerAggregationType = null;

// get the aggregation type of the record series category (for further use)
try
{
    parentRecordSeries.refresh();
    containerAggregationType =
parentRecordSeries.getProperties().get("AggregationType").getStringValue();
}
catch (Exception e)
{
    // if this property doesn't exist on the container, then this is not a valid
    category and the record shouldn't be managed by this algorithm
    return;
}

// if the record is filed into a sub-folder (and not a record series category),
get the KeyValue and KeyValueDescription of the sub-folder
if (parentFolder != null)
{
    try
    {
        parentFolder.refresh();
    }
}

```

```

        containerKeyValue =
parentFolder.getProperties().get("KeyValue").getStringValue();
    }
    catch (Exception e)
    {
        // if one of those properties doesn't exist on the parent folder, then this is
not a valid folder and the record shouldn't be managed by this algorithm
        return;
    }
}

// Check if the record is filed in the correct record series category by
comparing the current location with the one expected based on the source document
property
// If the record series has changed on the source document, then the record must be
relocated into a different record series category
    if (!containerRecordSeriesCode.equals(recordRecordSeriesCode))
    {
        relocateRecord(jarmRecord, recordRecordSeriesCode, recordKeyValue,
recordKeyValueDescription);
    }
    else
    {
        // If the record series container uses an aggregation record,
        // then the record is filed directly under it, and doesn't need to be moved to a
different subfolder.
        // Only maintain record location when filed into a subfolder
        if (containerAggregationType.equalsIgnoreCase("RecordFolder"))
        {
            if (containerKeyValue != null && recordKeyValue != null &&
!containerKeyValue.equals(recordKeyValue))
            {
                relocateRecord(jarmRecord, containerRecordSeriesCode, recordKeyValue,
recordKeyValueDescription);
            }
        }
    }
}

/*
 * Move the record to the new location
 */
private void relocateRecord(Record recordToRelocate, String recordSeriesCode, String
keyValue, String keyValueDescription) throws Exception
{
    RecordContainer destinationContainer = null;

    // Get the record series category which RecordCategoryIdentifier is equal to the
recordSeriesCode value
    RecordCategory recordSerieCategory =
returnRecordSeriesContainer(recordSeriesCode);
    String containerAggregationType =
recordSerieCategory.getProperties().get("AggregationType").getStringValue();
    if (containerAggregationType.equalsIgnoreCase("Record"))

```



```

        {
            destinationContainer = recordSerieCategory;
        }
        else
        {
            destinationContainer = returnRecordSeriesSubFolder(recordSerieCategory,
keyValue, keyValueDescription);
        }

        RecordContainer fromContainer = (RecordContainer)
recordToRelocate.getContainedBy().get(0);
        recordToRelocate.move(fromContainer, destinationContainer, "Record
relocated by event handler due to change in the document record serie code or the key
value");

        // For a Folder aggregation type, check if the source folder is empty after the
move, and if yes, delete it
        if (fromContainer instanceof RecordVolume &&
isContainerEmpty(fromContainer.getObjectIdentity()))
        {
            fromContainer.getParent().delete(false, DeleteMode.CheckRetainMetadata, "Folder
deleted");
        }
    }

    /*
    * Check if a folder is empty
    */
    private static boolean isContainerEmpty(String containerId)
    {
        // ---- Perform a search that returns ResultRows ----
        RMSearch jarmSearch = new RMSearch(jarmFPOS);
        // Check if the folder is associated with at least 1 record.
        String sqlStmt = "SELECT top 1 ContainmentName FROM
ReferentialContainmentRelationship r WHERE r.Tail = OBJECT(" + containerId + ")";
        // No object values returned.
        RMPPropertyFilter filter = null;
        // Not going to page.
        Integer pageSize = null;
        // Perform the search operation...
        Boolean continuable = Boolean.FALSE; // non-paging
        PageableSet<ResultRow> rowResultSet =
jarmSearch.fetchRows(sqlStmt, pageSize, filter, continuable);
        // Use normal iterator to access results in a non-paged manner.
        Iterator<ResultRow> iter = rowResultSet.iterator();
        return !iter.hasNext();
    }

    /* Illustrate a none paging search of a category
    * Search for a category with Record Category Identifier equals to the recordSerieId
parameter
    */
    private static RecordCategory returnRecordSeriesContainer(String recordSeriesCode)
throws Exception
    {

```

```

// ---- Perform a search that returns ResultRows ----
RMSearch jarmSearch = new RMSearch(jarmFPOS);
String sqlStmt = "SELECT rc.Id, rc.RecordCategoryName,
rc.RecordCategoryIdentifier FROM RecordCategory rc WHERE rc.RecordCategoryIdentifier =
'" + recordSeriesCode + "'";
// No object values returned.
RMPropertyFilter filter = null;
// Not going to page.
Integer pageSize = null;
// Perform the search operation...
Boolean continuable = Boolean.FALSE; // non-paging
PageableSet<ResultRow> rowResultSet =
jarmSearch.fetchRows(sqlStmt, pageSize, filter, continuable);
// Use normal iterator to access results in a non-paged manner.
Iterator<ResultRow> iter = rowResultSet.iterator();
int rowNumber = 0;
String rcId = null;
while ( iter.hasNext() )
{
    ResultRow row = iter.next();
    RMProperties rowProps = row.getProperties();
    rcId = rowProps.getStringValue("ID");
    rowNumber ++;
}

if (rowNumber != 1)
    throw new Exception("We are expecting to find one and only one category with
identifier = '" + recordSeriesCode + "', but we found " + rowNumber);
else
    return RMFactory.RecordCategory.fetchInstance(jarmFPOS, rcId, null);
}

/* Illustrate the getContent of a container
 * Search for a category with Record Category Identifier equals to the recordSerieId
parameter
 */

private static RecordFolder returnRecordSeriesSubFolder(RecordCategory
recordSerieCategory, String keyValue, String keyValueDescription) throws Exception
{
    RecordFolder subRecordFolder = null;

    // Build a filter to get the list of custom properties
    RMPropertyFilter filter = new RMPropertyFilter();
    Integer maxRecursion = Integer.valueOf(1); // Recursion depth to go down.
    Long maxContentSize = null; // No content involved - not applicable.
    Boolean levelDependents = null; // Use the repository default setting.
    Integer pageSize = null; // Use the repository default setting.
    // List of property symbolic names to include.
    String symbolicNames = "FolderName KeyValue KeyValueDescription";
    // Desired result set page size.

```

```

        filter.addIncludeProperty(maxRecursion, maxContentSize, levelDependents,
symbolicNames, pageSize);

        // Get all folders created under the record serie category
        // For demonstration purposes, use the paging mechanism of this PageableSet.
        PageableSet<RecordFolder> rcResultSet =
recordSerieCategory.fetchRecordFolders(filter, pageSize);
        RMPageIterator<RecordFolder> rcPI = rcResultSet.pageIterator();

        // Loop on all the pages
        while ( rcPI.nextPage() )
        {
            // For each page, loop on the folders
            List<RecordFolder> rcPage = rcPI.getCurrentPage();
            for (RecordFolder recFolder : rcPage)
            {
                if (keyValue.equals(recFolder.getProperties().getStringValue("KeyValue")))
                {
                    // Exit this look if we found the folder
                    subRecordFolder = recFolder;
                    break;
                }
            }

            // Exit the pagination loop if we have already found the folder
            if (subRecordFolder != null)
                break;
        }

        // If the specified sub folder has not been found under the record series
category, create it on the fly
        if (subRecordFolder == null)
        {
            RMPProperties jarmProps =
RMFactory.RMPProperties.createInstance(DomainType.P8_CE);
            jarmProps.putStringValue(RMPPropertyName.RecordFolderName, keyValueDescription
+ " - " + keyValue);
            jarmProps.putStringValue(RMPPropertyName.RecordFolderIdentifier, keyValue);
            jarmProps.putStringValue("keyValue", keyValue);
            jarmProps.putStringValue("KeyValueDescription", keyValueDescription);

            // Inherit the schedule from the parent container

            jarmProps.putObjectValue(com.ibm.jarm.api.constants.RMPPropertyName.DispositionSchedule,
recordSerieCategory.getAssignedSchedule());

            jarmProps.putGuidValue(com.ibm.jarm.api.constants.RMPPropertyName.DispositionScheduleInhe
ritedFrom, recordSerieCategory.getObjectIdentity());

            subRecordFolder = recordSerieCategory.addRecordFolder("ILGFolder", jarmProps,
null);
        }

        // Return the subfolder where to file the record to

```

```
        return subRecordFolder;
    }
}
```

Add the updated code into Content Engine event

To import those changes into the Content Engine events, follow these steps:

1. In Eclipse, generate a JAR file named `rm_autosyncproperties-handler.jar` to add to the `lib` folder. During the export process, select only the `AutoSyncProperties.java` file. Clear all other check boxes. The JAR file will contain only the `.class` compiled version of this file.
2. Update the `RMAutoSynchronizePropertiesImport.xml` file.

In the **<CodeModule>** section, search **<ContentElements>**. Add the code in Example 18-4 to the **<ContentTransfer>** section, as indicated in Figure 18-8 on page 409.

Example 18-4 Adding the necessary API references to the RMAutoSynchronizePropertiesImport.xml file

```
<ContentTransfer>
  <ObjectType>1038</ObjectType>
  <ContentType>application/x-zip-compressed</ContentType>
  <RetrievalName>Jarm.jar</RetrievalName>
  <ExternalRef>Jarm.jar</ExternalRef>
</ContentTransfer>
<ContentTransfer>
  <ObjectType>1038</ObjectType>
  <ContentType>application/x-zip-compressed</ContentType>
  <RetrievalName>JarmResources.jar</RetrievalName>
  <ExternalRef>JarmResources.jar</ExternalRef>
</ContentTransfer>
<ContentTransfer>
  <ObjectType>1038</ObjectType>
  <ContentType>application/x-zip-compressed</ContentType>
  <RetrievalName>ierLogTrace.jar</RetrievalName>
  <ExternalRef>ierLogTrace.jar</ExternalRef>
</ContentTransfer>
```

```

<ObjectManifest EMVersion="4.0">
  <Documents>
    - <CodeModule>
      - <CodeModuleProperties>
        <ObjectType>1</ObjectType>
        <Creator>p8admin</Creator>
        <DateCreated>2007-03-01T23:28:06.130Z</DateCreated>
        <LastModifier>p8admin</LastModifier>
        <DateLastModified>2007-03-01T23:28:06.130Z</DateLastModified>
        <Id>daa2e57d-e5e4-4dd4-9a6e-b63ac7a8a20c</Id>
        <Name>RMAutoSynchronizeProperties</Name>
        <SecurityPolicy />
        <SecurityParent />
        <IsFrozenVersion>0</IsFrozenVersion>
      + <VersionSeries>
        <MajorVersionNumber>1</MajorVersionNumber>
        <MinorVersionNumber>0</MinorVersionNumber>
        <VersionStatus>1</VersionStatus>
      - <ContentElements>
        - <ContentTransfer>
          <ObjectType>1038</ObjectType>
          <ContentType>application/x-zip-compressed</ContentType>
          <RetrievalName>rm_autosyncproperties-handler.jar</RetrievalName>
          <ExternalRef>rm_autosyncproperties-handler.jar</ExternalRef>
        </ContentTransfer>
        </ContentElements>
        <MimeType>application/x-zip-compressed</MimeType>

```

Figure 18-8 Adding the necessary API references to the RMAutoSynchronizePropertiesImport.xml file

3. Copy the entire project folder to the server where you run Enterprise Manager. Launch Enterprise Manager. Run the import scripts to create the event actions:
 - a. Right-click **Records Object Store**.
 - b. Select **Import All**.
 - c. In the Import Helper dialog, select the **RMAutoSynchronizePropertiesImport.xml** file.
 - d. For the External Content Path field, browse to the **lib** folder.
 - e. Click **Import**. At the end, you should see the "Import Success" notice. The import script creates the event action under the object store's Event Action folder and the code modules under the Code Module folder.
4. Subscribing to the RMAutoSynchronizeProperties event synchronizes the properties from the source document to its record.
 - a. In Enterprise Manager, navigate to **Object Stores/<ROS>/Document class**.

- b. Right-click and select **Add Subscription**.
- c. Specify a unique name and a description for the subscription.
- d. Select **Applies to all instances**.
- e. In the Specify Triggers window, select the event trigger named **Update**.
- f. Select the event action that you have imported.
- g. In the Specify Additional Properties window, check **Include Subclasses**.
- h. Click **Next** and **Save** to complete the process.

Related publications

The publications listed in this section are considered particularly suitable for more detailed information about the topics covered in this book.

IBM Redbooks

The following IBM Redbooks publications provide additional information about the topic in this document. Note that some publications referenced in this list might be available in softcopy only.

- ▶ *Working with IBM Records Manager*, SG24-7389
<http://www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/sg247389.html>
- ▶ *Quick Reference: Records Management 101*, TIPS0595
<http://www.redbooks.ibm.com/abstracts/tips0595.html>

You can search for, view, download or order these documents and other Redbooks, Redpapers, Web Docs, draft and additional materials, on the Redbooks web page:

ibm.com/redbooks

Online resources

These websites are also relevant as further information sources:

- ▶ IBM Enterprise Records publication library
<http://www.ibm.com/support/docview.wss?uid=swg27042282>
This website contains product release information, product documentation, and technical notices.
- ▶ IBM Enterprise Content Management
<http://ibm.co/1KHCuxH>
- ▶ IBM Content Collector information
<http://www.ibm.com/software/data/content-management/content-collector>

- ▶ IBM Watson Content Analytics
<http://www.ibm.com/software/products/en/watson-content-analytics>
- ▶ IBM FileNet P8 Platform
<http://www.ibm.com/software/data/content-management/filenet-p8-platform>
- ▶ IBM FileNet Content Manager
<http://www.ibm.com/software/products/en/filecontmana>

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Redbooks

Using IBM Enterprise Records

SG24-7623-01

ISBN 073844071X



(0.5" spine)

0.475" <-> 0.873"

250 <-> 459 pages



SG24-7623-01

ISBN 073844071X

Printed in U.S.A.

Get connected

